

RECEIVED

MAY 8 - 2001

12

# EUROPEAN PATENT APPLICATION

21 Application number: 88301052.2

51 Int. Cl.<sup>5</sup>: G06F 15/20 Technology Center 2100

22 Date of filing: 09.02.88

30 Priority: 13.02.87 US 14853

71 Applicant: Fischer, Addison M.  
60 14th Avenue South  
Naples Florida 33940(US)

43 Date of publication of application:  
17.08.88 Bulletin 88/33

72 Inventor: Fischer, Addison M.  
60 14th Avenue South  
Naples Florida 33940(US)

84 Designated Contracting-States:  
AT BE CH DE ES FR GB GR IT LI LU NL SE

88 Date of deferred publication of the search report:  
04.07.90 Bulletin 90/27

74 Representative: Enskat, Michael Antony Frank  
et al  
Saunders & Dolleymore 9, Rickmansworth  
Road  
Watford Hertfordshire WD1 7HE(GB)

54 Document composition system using named formats and named fonts.

57 An interactive data processing implemented method and apparatus for composing and editing a document in which a user is afforded great flexibility in defining the document geometry and in changing the data presentation characteristics associated with non-contiguous portions of the document. According to the present invention, each of the distinct line formats in the document is assigned an abstract format name (i.e., a Named Format) and each line in the document is associated with a distinct format (See A, Q, D, H, Figure 2). A Named Format is associated with each fragment of the text in the document. Likewise, an abstract name, referred to as

a Named Font, is associated with a wide range of specifiable data presentation characteristics which operate over a range of the document defined by the user (see Figure 4). The same Named Format and Named Font is used to simultaneously specify different data presentation characteristics for the output devices (2, 4, 6, 8, 12, 20) of the system (see Figure 3). By a single change in the specification for a Named Format (Figures 2, 3) or Named Font (Figure 4), all the underlying data throughout the text which is associated with a Named Format or Font is automatically changed to reflect the modified characteristic specification.

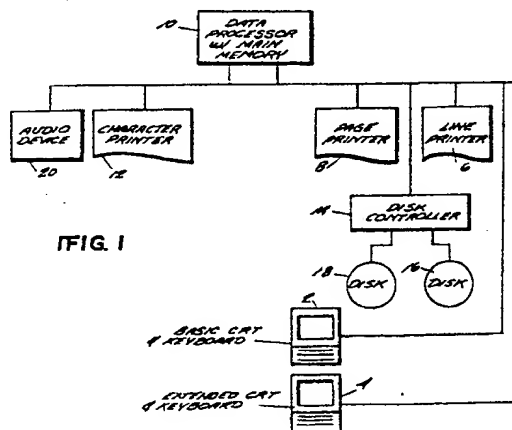


FIG. 1

EP 0 278 722 A3



European Patent  
Office

# EUROPEAN SEARCH REPORT

Application Number

EP 88 30 1052

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.4)
X	COMMUNICATIONS OF THE ASSOCIATION OF COMPUTING MACHINERY, vol. 28, no. 9, September 1985, pages 942-960, ACM, New York, N.Y., US; J. GUTKNECHT: "Concepts of the text editor LARA" * Page 943, right-hand column, paragraph 5 - page 944, right-hand column, paragraph 2; page 948, right-hand column, paragraph 2 - page 953, right-hand column, paragraph 1 *	1	G 06 F 15/20
A	IDEM ---	2,4	
X	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 27, no. 10B, March 1985, page 5985; "Providing three kinds of attributes for object data and properties" * Whole document *	1	
A	EP-A-0 136 711 (WANG) * Page 4, line 1 - page 5, line 24; page 7, lines 3-25; page 38, line 14 - page 43, line 3; page 61, line 6 - page 63, line 16; figure 4 *	1,2,4-6	
			TECHNICAL FIELDS SEARCHED (Int. Cl.4)
			G 06 F 15/20
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 28-03-1990	Examiner CHUGG D.J.
<b>CATEGORY OF CITED DOCUMENTS</b>			
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons ----- & : member of the same patent family, corresponding document	

(12)

**EUROPEAN PATENT APPLICATION**

(21) Application number: **88301052.2**

(51) Int. Cl.<sup>4</sup>: **G 06 F 15/20**

(22) Date of filing: **09.02.88**

**RECEIVED**  
**MAY 8 - 2001**  
Technology Center 2100

(30) Priority: **13.02.87 US 14853**

(43) Date of publication of application:  
**17.08.88 Bulletin 88/33**

(84) Designated Contracting States:  
**AT BE CH DE ES FR GB GR IT LI LU NL SE**

(71) Applicant: **Fischer, Addison M.**  
**60 14th Avenue South**  
**Naples Florida 33940 (US)**

(72) Inventor: **Fischer, Addison M.**  
**60 14th Avenue South**  
**Naples Florida 33940 (US)**

(74) Representative: **Enskat, Michael Antony Frank et al**  
**Saunders & Dolleymore 9, Rickmansworth Road**  
**Watford Hertfordshire WD1 7HE (GB)**

**(54) Document composition system using named formats and named fonts.**

(57) An interactive data processing implemented method and apparatus for composing and editing a document in which a user is afforded great flexibility in defining the document geometry and in changing the data presentation characteristics associated with non-contiguous portions of the document. According to the present invention, each of the distinct line formats in the document is assigned an abstract format name (i.e., a Named Format) and each line in the document is associated with a distinct format (See A, Q, D, H, Figure 2). A Named Format is associated with each fragment of the text in the document. Likewise, an abstract name, referred to as a Named Font, is associated with a wide range of specifiable data presentation characteristics which operate over a range of the document defined by the user (see Figure 4). The same Named Format and Named Font is used to simultaneously specify different data presentation characteristics for the output devices (2, 4, 6, 8, 12, 20) of the system (see Figure 3). By a single change in the specification for a Named Format (Figures 2, 3) or Named Font (Figure 4), all the underlying data throughout the text which is associated with a Named Format or Font is automatically changed to reflect the modified characteristic specification.

**DETAILED SPECIFICATIONS FOR FONT ID = BEN**

BASIC CRT	EXTENDED CRT	LINE PRT	PAGE PRT
HEIGHT — 100% — 120% — 140% — 160% — 180% — 200% — 220% — 240% — 260% — 280% — 300% — 320% — 340% — 360% — 380% — 400% — 420% — 440% — 460% — 480% — 500% — 520% — 540% — 560% — 580% — 600% — 620% — 640% — 660% — 680% — 700% — 720% — 740% — 760% — 780% — 800% — 820% — 840% — 860% — 880% — 900% — 920% — 940% — 960% — 980% — 1000% — 1020% — 1040% — 1060% — 1080% — 1100% — 1120% — 1140% — 1160% — 1180% — 1200% — 1220% — 1240% — 1260% — 1280% — 1300% — 1320% — 1340% — 1360% — 1380% — 1400% — 1420% — 1440% — 1460% — 1480% — 1500% — 1520% — 1540% — 1560% — 1580% — 1600% — 1620% — 1640% — 1660% — 1680% — 1700% — 1720% — 1740% — 1760% — 1780% — 1800% — 1820% — 1840% — 1860% — 1880% — 1900% — 1920% — 1940% — 1960% — 1980% — 2000% — 2020% — 2040% — 2060% — 2080% — 2100% — 2120% — 2140% — 2160% — 2180% — 2200% — 2220% — 2240% — 2260% — 2280% — 2300% — 2320% — 2340% — 2360% — 2380% — 2400% — 2420% — 2440% — 2460% — 2480% — 2500% — 2520% — 2540% — 2560% — 2580% — 2600% — 2620% — 2640% — 2660% — 2680% — 2700% — 2720% — 2740% — 2760% — 2780% — 2800% — 2820% — 2840% — 2860% — 2880% — 2900% — 2920% — 2940% — 2960% — 2980% — 3000% — 3020% — 3040% — 3060% — 3080% — 3100% — 3120% — 3140% — 3160% — 3180% — 3200% — 3220% — 3240% — 3260% — 3280% — 3300% — 3320% — 3340% — 3360% — 3380% — 3400% — 3420% — 3440% — 3460% — 3480% — 3500% — 3520% — 3540% — 3560% — 3580% — 3600% — 3620% — 3640% — 3660% — 3680% — 3700% — 3720% — 3740% — 3760% — 3780% — 3800% — 3820% — 3840% — 3860% — 3880% — 3900% — 3920% — 3940% — 3960% — 3980% — 4000% — 4020% — 4040% — 4060% — 4080% — 4100% — 4120% — 4140% — 4160% — 4180% — 4200% — 4220% — 4240% — 4260% — 4280% — 4300% — 4320% — 4340% — 4360% — 4380% — 4400% — 4420% — 4440% — 4460% — 4480% — 4500% — 4520% — 4540% — 4560% — 4580% — 4600% — 4620% — 4640% — 4660% — 4680% — 4700% — 4720% — 4740% — 4760% — 4780% — 4800% — 4820% — 4840% — 4860% — 4880% — 4900% — 4920% — 4940% — 4960% — 4980% — 5000% — 5020% — 5040% — 5060% — 5080% — 5100% — 5120% — 5140% — 5160% — 5180% — 5200% — 5220% — 5240% — 5260% — 5280% — 5300% — 5320% — 5340% — 5360% — 5380% — 5400% — 5420% — 5440% — 5460% — 5480% — 5500% — 5520% — 5540% — 5560% — 5580% — 5600% — 5620% — 5640% — 5660% — 5680% — 5700% — 5720% — 5740% — 5760% — 5780% — 5800% — 5820% — 5840% — 5860% — 5880% — 5900% — 5920% — 5940% — 5960% — 5980% — 6000% — 6020% — 6040% — 6060% — 6080% — 6100% — 6120% — 6140% — 6160% — 6180% — 6200% — 6220% — 6240% — 6260% — 6280% — 6300% — 6320% — 6340% — 6360% — 6380% — 6400% — 6420% — 6440% — 6460% — 6480% — 6500% — 6520% — 6540% — 6560% — 6580% — 6600% — 6620% — 6640% — 6660% — 6680% — 6700% — 6720% — 6740% — 6760% — 6780% — 6800% — 6820% — 6840% — 6860% — 6880% — 6900% — 6920% — 6940% — 6960% — 6980% — 7000% — 7020% — 7040% — 7060% — 7080% — 7100% — 7120% — 7140% — 7160% — 7180% — 7200% — 7220% — 7240% — 7260% — 7280% — 7300% — 7320% — 7340% — 7360% — 7380% — 7400% — 7420% — 7440% — 7460% — 7480% — 7500% — 7520% — 7540% — 7560% — 7580% — 7600% — 7620% — 7640% — 7660% — 7680% — 7700% — 7720% — 7740% — 7760% — 7780% — 7800% — 7820% — 7840% — 7860% — 7880% — 7900% — 7920% — 7940% — 7960% — 7980% — 8000% — 8020% — 8040% — 8060% — 8080% — 8100% — 8120% — 8140% — 8160% — 8180% — 8200% — 8220% — 8240% — 8260% — 8280% — 8300% — 8320% — 8340% — 8360% — 8380% — 8400% — 8420% — 8440% — 8460% — 8480% — 8500% — 8520% — 8540% — 8560% — 8580% — 8600% — 8620% — 8640% — 8660% — 8680% — 8700% — 8720% — 8740% — 8760% — 8780% — 8800% — 8820% — 8840% — 8860% — 8880% — 8900% — 8920% — 8940% — 8960% — 8980% — 9000% — 9020% — 9040% — 9060% — 9080% — 9100% — 9120% — 9140% — 9160% — 9180% — 9200% — 9220% — 9240% — 9260% — 9280% — 9300% — 9320% — 9340% — 9360% — 9380% — 9400% — 9420% — 9440% — 9460% — 9480% — 9500% — 9520% — 9540% — 9560% — 9580% — 9600% — 9620% — 9640% — 9660% — 9680% — 9700% — 9720% — 9740% — 9760% — 9780% — 9800% — 9820% — 9840% — 9860% — 9880% — 9900% — 9920% — 9940% — 9960% — 9980% — 10000% — 10020% — 10040% — 10060% — 10080% — 10100% — 10120% — 10140% — 10160% — 10180% — 10200% — 10220% — 10240% — 10260% — 10280% — 10300% — 10320% — 10340% — 10360% — 10380% — 10400% — 10420% — 10440% — 10460% — 10480% — 10500% — 10520% — 10540% — 10560% — 10580% — 10600% — 10620% — 10640% — 10660% — 10680% — 10700% — 10720% — 10740% — 10760% — 10780% — 10800% — 10820% — 10840% — 10860% — 10880% — 10900% — 10920% — 10940% — 10960% — 10980% — 11000% — 11020% — 11040% — 11060% — 11080% — 11100% — 11120% — 11140% — 11160% — 11180% — 11200% — 11220% — 11240% — 11260% — 11280% — 11300% — 11320% — 11340% — 11360% — 11380% — 11400% — 11420% — 11440% — 11460% — 11480% — 11500% — 11520% — 11540% — 11560% — 11580% — 11600% — 11620% — 11640% — 11660% — 11680% — 11700% — 11720% — 11740% — 11760% — 11780% — 11800% — 11820% — 11840% — 11860% — 11880% — 11900% — 11920% — 11940% — 11960% — 11980% — 12000% — 12020% — 12040% — 12060% — 12080% — 12100% — 12120% — 12140% — 12160% — 12180% — 12200% — 12220% — 12240% — 12260% — 12280% — 12300% — 12320% — 12340% — 12360% — 12380% — 12400% — 12420% — 12440% — 12460% — 12480% — 12500% — 12520% — 12540% — 12560% — 12580% — 12600% — 12620% — 12640% — 12660% — 12680% — 12700% — 12720% — 12740% — 12760% — 12780% — 12800% — 12820% — 12840% — 12860% — 12880% — 12900% — 12920% — 12940% — 12960% — 12980% — 13000% — 13020% — 13040% — 13060% — 13080% — 13100% — 13120% — 13140% — 13160% — 13180% — 13200% — 13220% — 13240% — 13260% — 13280% — 13300% — 13320% — 13340% — 13360% — 13380% — 13400% — 13420% — 13440% — 13460% — 13480% — 13500% — 13520% — 13540% — 13560% — 13580% — 13600% — 13620% — 13640% — 13660% — 13680% — 13700% — 13720% — 13740% — 13760% — 13780% — 13800% — 13820% — 13840% — 13860% — 13880% — 13900% — 13920% — 13940% — 13960% — 13980% — 14000% — 14020% — 14040% — 14060% — 14080% — 14100% — 14120% — 14140% — 14160% — 14180% — 14200% — 14220% — 14240% — 14260% — 14280% — 14300% — 14320% — 14340% — 14360% — 14380% — 14400% — 14420% — 14440% — 14460% — 14480% — 14500% — 14520% — 14540% — 14560% — 14580% — 14600% — 14620% — 14640% — 14660% — 14680% — 14700% — 14720% — 14740% — 14760% — 14780% — 14800% — 14820% — 14840% — 14860% — 14880% — 14900% — 14920% — 14940% — 14960% — 14980% — 15000% — 15020% — 15040% — 15060% — 15080% — 15100% — 15120% — 15140% — 15160% — 15180% — 15200% — 15220% — 15240% — 15260% — 15280% — 15300% — 15320% — 15340% — 15360% — 15380% — 15400% — 15420% — 15440% — 15460% — 15480% — 15500% — 15520% — 15540% — 15560% — 15580% — 15600% — 15620% — 15640% — 15660% — 15680% — 15700% — 15720% — 15740% — 15760% — 15780% — 15800% — 15820% — 15840% — 15860% — 15880% — 15900% — 15920% — 15940% — 15960% — 15980% — 16000% — 16020% — 16040% — 16060% — 16080% — 16100% — 16120% — 16140% — 16160% — 16180% — 16200% — 16220% — 16240% — 16260% — 16280% — 16300% — 16320% — 16340% — 16360% — 16380% — 16400% — 16420% — 16440% — 16460% — 16480% — 16500% — 16520% — 16540% — 16560% — 16580% — 16600% — 16620% — 16640% — 16660% — 16680% — 16700% — 16720% — 16740% — 16760% — 16780% — 16800% — 16820% — 16840% — 16860% — 16880% — 16900% — 16920% — 16940% — 16960% — 16980% — 17000% — 17020% — 17040% — 17060% — 17080% — 17100% — 17120% — 17140% — 17160% — 17180% — 17200% — 17220% — 17240% — 17260% — 17280% — 17300% — 17320% — 17340% — 17360% — 17380% — 17400% — 17420% — 17440% — 17460% — 17480% — 17500% — 17520% — 17540% — 17560% — 17580% — 17600% — 17620% — 17640% — 17660% — 17680% — 17700% — 17720% — 17740% — 17760% — 17780% — 17800% — 17820% — 17840% — 17860% — 17880% — 17900% — 17920% — 17940% — 17960% — 17980% — 18000% — 18020% — 18040% — 18060% — 18080% — 18100% — 18120% — 18140% — 18160% — 18180% — 18200% — 18220% — 18240% — 18260% — 18280% — 18300% — 18320% — 18340% — 18360% — 18380% — 18400% — 18420% — 18440% — 18460% — 18480% — 18500% — 18520% — 18540% — 18560% — 18580% — 18600% — 18620% — 18640% — 18660% — 18680% — 18700% — 18720% — 18740% — 18760% — 18780% — 18800% — 18820% — 18840% — 18860% — 18880% — 18900% — 18920% — 18940% — 18960% — 18980% — 19000% — 19020% — 19040% — 19060% — 19080% — 19100% — 19120% — 19140% — 19160% — 19180% — 19200% — 19220% — 19240% — 19260% — 19280% — 19300% — 19320% — 19340% — 19360% — 19380% — 19400% — 19420% — 19440% — 19460% — 19480% — 19500% — 19520% — 19540% — 19560% — 19580% — 19600% — 19620% — 19640% — 19660% — 19680% — 19700% — 19720% — 19740% — 19760% — 19780% — 19800% — 19820% — 19840% — 19860% — 19880% — 19900% — 19920% — 19940% — 19960% — 19980% — 20000% — 20020% — 20040% — 20060% — 20080% — 20100% — 20120% — 20140% — 20160% — 20180% — 20200% — 20220% — 20240% — 20260% — 20280% — 20300% — 20320% — 20340% — 20360% — 20380% — 20400% — 20420% — 20440% — 20460% — 20480% — 20500% — 20520% — 20540% — 20560% — 20580% — 20600% — 20620% — 20640% — 20660% — 20680% — 20700% — 20720% — 20740% — 20760% — 20780% — 20800% — 20820% — 20840% — 20860% — 20880% — 20900% — 20920% — 20940% — 20960% — 20980% — 21000% — 21020% — 21040% — 21060% — 21080% — 21100% — 21120% — 21140% — 21160% — 21180% — 21200% — 21220% — 21240% — 21260% — 21280% — 21300% — 21320% — 21340% — 21360% — 21380% — 21400% — 21420% — 21440% — 21460% — 21480% — 21500% — 21520% — 21540% — 21560% — 21580% — 21600% — 21620% — 21640% — 21660% — 21680% — 21700% — 21720% — 21740% — 21760% — 21780% — 21800% — 21820% — 21840% — 21860% — 21880% — 21900% — 21920% — 21940% — 21960% — 21980% — 22000% — 22020% — 22040% — 22060% — 22080% — 22100% — 22120% — 22140% — 22160% — 22180% — 22200% — 22220% — 22240% — 22260% — 22280% — 22300% — 22320% — 22340% — 22360% — 22380% — 22400% — 22420% — 22440% — 22460% — 22480% — 22500% — 22520% — 22540% — 22560% — 22580% — 22600% — 22620% — 22640% — 22660% — 22680% — 22700% — 22720% — 22740% — 22760% — 22780% — 22800% — 22820% — 22840% — 22860% — 22880% — 22900% — 22920% — 22940% — 22960% — 22980% — 23000% — 23020% — 23040% — 23060% — 23080% — 23100% — 23120% — 23140% — 23160% — 23180% — 23200% — 23220% — 23240% — 23260% — 23280% — 23300% — 23320% — 23340% — 23360% — 23380% — 23400% — 23420% — 23440% — 23460% — 23480% — 23500%			

## Description

### DOCUMENT COMPOSITION SYSTEM USING NAMED FORMATS AND NAMED FONTS

#### FIELD OF THE INVENTION

This invention relates to an interactive data processing implemented method and apparatus for composing and editing a document, especially in an office or business environment. More particularly, this invention relates to a word processing system in which the user is afforded great flexibility in defining the document geometry and in changing the data presentation characteristics associated with non-contiguous portions of the document.

#### BACKGROUND OF THE INVENTION

There are available word processors, such as the Macintosh, which allow the user to define a "ruler" in which document geometry characteristics such as a left margin, right margin and indentation point may be defined. The "rulers" in such prior art word processing systems have heretofore been associated only with contiguous groups of lines. There is no easy way to change "rulers" throughout the entire document. This is particularly a problem when composing a lengthy document in which it is necessary to switch back and forth many times between interleaved rulers.

The IBM Script document composition facility allows margins and columns to be changed throughout a document. However, in order to even permit the possibility of changes throughout a document, the Script system requires the tedious and labor intensive process of embedding commands throughout the document. The need to embed commands throughout a document in the IBM Script system, makes document composition extremely tedious and very error prone, often requiring many iterations to debug the errors in the embedded commands.

Additionally, the IBM Script system precludes the user from seeing, in a form approaching final form, the document as it is being created because of the clutter due to embedded commands. The original document typically becomes very difficult to follow due to the embedded, cryptic commands which break up the actual textual material.

In prior art word processing systems, changing a ruler does not retroactively change the lines which were entered under the ruler. In many prior art word processing systems, only by changing the ruler can a differently formatted line be entered at all. Typically, once a line is entered any reformatting requires attention to the individual line.

In prior art word processing editors, the document being produced on disparate output devices has to be individually formatted for each device, if such capability is present at all. No prior art technique exists for automatically formatting of a document in parallel on different document processing output devices without requiring, at a minimum, the use of formatting commands which usually appear embedded within the document during its composition. In prior art document composition facilities, any

capability for automatically formatting a document, in parallel, for different output devices has required the insertion of instructions throughout the document during its composition -- which renders composition slow, tedious, painstaking and prone to errors.

#### OBJECTS OF THE INVENTION

An object of the present invention is to provide a document composition word processing system which has highly flexible document revision capabilities and which has with the maximum degree of user friendliness.

A further object of the invention is to create documents that look very close to final form as they are being created (i.e., without being cluttered by embedded commands) and which are immediately transposable into final formatted form for each of the system's document producing output devices.

A still further object of the present invention is to provide a document composition system in which non-contiguous portions of a document may be readily reformatted or otherwise modified, by merely changing a single data presentation characteristic specification. The present invention requires no laborious process of embedding commands or other coded specifications throughout the document in order to make changes throughout the document (as does IBM Script Document Composition Facility). Moreover, the present invention allows the user to see the document as it is being created in a form approaching final form, (i.e., not cluttered with embedded commands).

A still further object of the present invention is to simultaneously define distinct formats and data presentation characteristics for disparate document producing output devices. The present invention readily permits the automatic reformatting of a document on these different output devices including the output device used during the process of composition.

Other advantages and objects of this invention will be more fully appreciated from the description of the invention which follows.

#### SUMMARY OF THE INVENTION

The present invention evolved in part from the recognition that in a given business document there is a need for a number of distinct line formats. Each format defines one of an essentially infinite number of geometric and/or topological arrangements of alphanumeric textual data which a user might find desirable for some portion of the document. For example, the main body of text in a document may have one main format having predetermined left and right margins, paragraph indentations and tabs. However, in addition the document may include several sections having auxiliary materials, such as embedded quotes, as well as sections where there are one or more columns of numeric data. Such text portions may be interleaved.

In the present invention, each of these distinct

formats in the text is assigned an abstract format name and each line (or paragraph or any other fragment of text) in the document is associated with the Format Name. The lines which are not associated with a particular format by the user are automatically assigned a "default" format by the system. Thus, an abstract format name is associated with each fragment of text in the document. Throughout this text, the abstract format name will be referred to as a Named format.

Once a name has been associated with each line of text, that name can simultaneously define a parallel set of definitions for each of a predetermined number of output devices of the system. For example, a Named Format could simultaneously specify a display window width spanning 76 characters for a particular interactive display device and a 133 character width line for a particular line printer. With respect to a laser printer, the same Named Format may have yet another line specifying connotation, perhaps indicating dimensions in inches, millimeters or any other convenient scale.

In the present invention, a Named Format may be changed by the user to, for example, change the specifications for the left and right margins of a single field format, or changing the widths of columns in a multi-field format. Once such changes have been made, the data associated with the respective formats will be automatically reformatted to correspond with the assigned Named Format. Thus, all data associated with a given Named Format which is interleaved with other formats throughout the document may be reformatted after changing only a single format specification.

A Named Format field can also specify other attributes and checks for the data such as: spell checking, numeric field validation and other data checking functions by passing control to various system editing routines. In addition, each field or fields in a Named Format is tied to or associated with an additional set of data presentation characteristics, by a mechanism hereinafter referred to as a Named Font.

The use of a Named Font is not, however, limited to defining characteristics of a Format field. For the purposes of this text, it must be emphasized that the term "Named Font" is used in a much broader sense than the common use of the term (i.e., to merely define a type style). As used hereinafter, the term "Font" broadly refers to data presentation characteristics. For example, each Named Font will specify data presentation characteristics including one or more of display intensity, flashing, displayed underscore, reverse video, color, character set, type style (e.g., Roman Times, Serif, etc.), type modifier (e.g., italics, boldface, etc.), point size, etc. In addition, a Named Font may be used to define the particular parts of a document to be delivered in audio even indicating the volume, gender, accent or any other available selectable data presentation characteristic for an audio device.

Each Named Format field, or a group of Named Format fields may be associated with a Named Font. Alternatively, Named Fonts may be defined independently throughout the text of a document where

the Named Font range is variable and user specified. Named Fonts may also be used to indicate fragments of text to be treated as footnotes, index or table of contents material or the like.

Like a Named Format, a single Named Font may take on different data presentations meanings for each of the output devices in the system. For example, a Font named "Apple" may specify a display in red on an extended function CRT, a printout in italics on a page printer, and an underlined printout on a line printer. If such a Named Font were to be associated with the first field of a Named Format "Y", then everywhere in the text in which format "Y" appears, the first field would take on the above-mentioned characteristics (unless some or all of the characteristics are subsequently respecified by the user) and a final document would be produced having red, italics, and underlined presentations on an extended function CRT, page printer and a line printer, respectively.

It is important to note that the Named Format and Fonts are associated with the data implicitly, rather than being explicitly indicated by embedded specifications or control characters which would be visible during composition.

One of the many advantages of the present is that it provides the user with a flexible and efficient mechanism for simultaneously utilizing the highlighting features distinctive to each particular device on which the document is displayed or produced. This includes the CRTs on which the document is composed (where highlighting capabilities include: foreground color, background color, reverse video, flashing, underscoring, outlining, intensity). The highlighting features of such non-interactive output devices as line printers (where highlighting characteristics may include: darkness, underscoring and other overstriking) page printers (where highlighting characteristics often include: type style, print size, italics, boldface, underscoring, outlining, etc.), and other devices, including, but not limited to, audio devices (where highlighting characteristics might include: decibel level, voice accent, voice guarder, and speed of speech) are also advantageously utilized.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is an exemplary data processing system in which the present invention may be implemented.

Figure 2 is an exemplary display screen depicting four Named Formats.

Figure 3 is an exemplary display screen from which a user may view and specify details for the Named Format A of Figure 2.

Figure 4 is an exemplary display screen from which a user may detail specifications for a Named Font.

Figures 5A, 5B, 6A and 6B are flowcharts showing an exemplary sequence of operations involved in the use of multiple Named Fonts for the same portion of a document.

Figure 7 is an overall block diagram of the subsystems which control the data processing system of the present invention.

Figure 8 shows the structure of a typical document record.

Figure 9 generally shows the fields in a point manager element.

Figures 10A, B, C and D are flowcharts showing operations performed by the file manager.

Figure 11 shows linked point manager element blocks of the point manager.

Figure 12 is a flowchart showing how the point manager creates a point block.

Figures 13-15 are flowcharts showing the basic sequence of operations performed by the display and output manager.

Figures 16-17 are flowcharts showing the basic sequence of operations performed by the input and command manager.

#### DETAILED DESCRIPTION

The present invention may be implemented in a wide range of different data processing systems. It is emphasized that the data processing components described with respect to Figure 1 are mentioned by way of example only and should not be construed as limiting the practice of the present invention in any way.

The CRT 2 may, for example, be an IBM 3270 monochrome display terminal, or a color 3279 display, or an IBM personal computer display. The extended function CRT 4 may, for example, be the IBM 3279 terminal which supports color graphics. Each of these CRT terminals is associated with a keyboard for entering data and commands. It should be recognized that when the term "CRT" is used herein, any functionally equivalent interactive display device which can be interfaced with a data processing system could be substituted therefor. Such equivalent devices include the general class of so-called computer "display" devices. For example, plasma-screen devices as well as teletype-type devices fall into this category.

The data processor 10, which is coupled to CRT 2 and CRT 4, is generally shown with its main memory and may be any of a number of general purpose digital computers. For example, the data processor 10 may be an IBM 370-equivalent type processor such as the IBM 4341, model 2 with its associated operating system. Alternatively, the processor may be a PC or any other general purpose processor that supports display screens. As will be discussed in detail below, processor 10 processes entered document data and commands received from either of the CRT keyboards. In addition, data processor 10 executes the programs resident in its memory so as to control the operation of the document processing system of the present invention.

Turning next to the printers shown in Figure 1, the line printer 6 may, for example, be the IBM 3203, model 5 Line Printer, and the page printer 8 may, for example, be an IBM 4250 Electro Erosion page printer. The printer 8, which produces equivalent results to a laser printer, may alternatively be a laser printer. The system may also include a character printer 12.

The system also includes auxiliary storage devices

such as disc storage devices 16, 18 which are controlled by controller 14.

In addition, the system may include an audio device 20, which may, for example, be Digital Equipment DEC TALK. This device typically receives a digital signal from data processor 10 and thereafter accesses its stored vocabulary of words to drive a speech synthesizer to vocalize a desired message.

#### DOCUMENT COMPOSITION

The nature of the Named Formats and Named Fonts of the present invention is best understood by focusing on the manner in which a document would typically be composed on the system. In order not to obscure the present invention, those aspects of document composition and editing which are shared with prior art word processing systems will not be described in detail, but will be readily understood by those of ordinary skill in the art.

To compose a document a user will typically begin by calling Named Formats and Fonts from the system library or by creating and detailing his own. The system's default attribute will supply "default" Formats and Fonts if none are defined by the user.

The Named Format defines the physical and geometric or topological structure of the document. Each Named Format has one or more fields and each field has associated therewith a Named Font. The Named Font is used to associate additional data presentation characteristics with the Named Format field. It should be recognized, however, that such additional characteristics could have been specified directly in the Named Format. The named Formats and Fonts control how the text in the associated line is to be processed and presented.

In practice, a user may initially define either a Named Font or Named Format. Typically, a user will begin preparation of a document by defining Named Formats which set forth the global geometry of the document, e.g., page width, the left and right margins, tab(s) for paragraph indenting, the manner in which pages are to begin and end, whether the data is to be left and/or right justified, or centered. In addition, in order to override the aforementioned "default" system, the user may define a Named Format and Font which will be controlling, if no format or font is otherwise specified.

To prompt the user to set up all the aspects of the document relating to its overall geometry, the keyboards associated with CRT 2 and 4 may be provided with a DEFINE DOCUMENT key. By depressing this key, the system will prompt the user to define "Global Document" formats. The user may at this point choose to define his own "default" format which the system will revert to if no other is specified. Additionally, the user will typically define special formats to be automatically used for page top and bottom, if desired. To define special formats to be automatically used, the user will press a DISPLAY FORMAT program function key, or otherwise execute a command which will result in the display at the user's CRT of the Named Format such as formats A, Q, D and H shown in Figure 2.

The user may then decide that, for example, Format A is the basic format defining the geometric

layout for the bulk of his document. The user moves the cursor to initially set a specific left and right margin, and paragraph indentation for Format A for the user's CRT.

The user makes this selection by typing, for example, an S (Start), I and E (end) on the ruler associated with Format A. By typing in L (left justified) and R (right justified) instead of S and E, the user can effect left or right justification. In an exemplary embodiment of the present invention, the Format fields of the CRT may be specified quickly by over typing characters on a prototype line. S and L are used to denote the left margin for a column of a particular field. L and S are similar, both define the left hand side of the column of the field, except that L implies the data should be left justified. E and R are similar in that both define the right hand side of the column of the field except that R implies the data should be right justified. I indicates the "Indentation" point when the format is associated with the first line of a paragraph.

The user might then execute a command called "DETAIL FORMAT" to allow a detailed description for Named Format A to be defined as shown in Figure 3. The user will first turn to field 1 of format A. It is noted that most Formats for textual material will only have one field. However, multiple fields are especially useful when creating parallel columns of numeric data. For field one, the user will have entries to make for each of the systems' output devices, e.g., a basic function CRT, an extended function CRT, a line printer, a graphics, page, or laser printer, etc. As will be apparent to those skilled in the art, although the details of a Named Format or Named Font are specified using menus, commands could have alternatively been utilized to accomplish the same result.

Additionally, the user will have a Font ID field and a "characteristics" to fill in. In regard to the Font ID field, the user may associate a Font named GEORGE with this field to define the normal color type face or other attributes to be associated with data in this field. The "characteristics" field may, for example, be utilized to specify left justified, right justified, hyphenated and word wrapped (i.e., adjustment of the data on entry if it exceeds a pre-defined boundary characteristic) as well as "spell checking", "numeric only", or any of a variety of other data checks and verification criteria. As shown in Figure 3, these characteristics could be selected by the user by entering, for example, the code "L, R, H, W" and, for example, "SP" for spellcheck.

As shown in Figure 3, the Named Format may also be used to simultaneously define how the associated text may be displayed on the system output devices. The user may, for example, specify a format field of 1-69 characters for a basic CRT, 1-120 characters for an extended CRT, 1-132 characters for a line printer and a page width of .5-8.0 inches for a laser, page, or graphics printer.

The Named Font GEORGE can be used to allow the user to specifically select further data presentation attributes to be normally associated with this field as will be described below. Alternatively, the user may choose not to identify a Font for this field,

in which case the system will supply a default Font.

Although only one field is contained in exemplary Format A, it should be recognized that the Format may have additional fields in which similar data may be entered. The user may then define detailed specifications for Formats setting up how columns of numbers are to be displayed, such as shown in Format D (which has 3 fields), or embedded auxiliary (e.g., quoted) material as shown by Format Q in Figure 2. For multiple field formats such as Format D, the start and end of each field in the format may, for example, be denoted by S and E respectively.

Additionally, the user may define a global format defining the geometry of a page, document or document section header such as Format H shown in Figure 2. Likewise, a page footer may be defined showing the desired format for ending a page (e.g., by a centered number, etc.) to be automatically used to format the footers of the pages within the document.

After the user has defined the basic document formats, he may optionally change the specifications of any or all the formats by striking a MODIFY FORMAT key or otherwise directing a modify format command. After the document has been stored and is in the process of being edited, if the specification of format A, for example, is changed, then the entire file for the document associated with Format A will be modified according to the revised rules. Thus, if the user changes the left margin setting for Format A, the left margin of all data lines associated with Format A will automatically be changed. Similar changes may likewise be made with respect to the Named Fonts with a similar modifying effect. If the associated changes to a Named Format or Font affect an aspect of CRT display, then the document will be immediately reformatted.

The user may next define the Named Fonts for his document. To define a Font the user will strike a DEFINE FONT key on the keyboard associated with the CRT. An exemplary display resulting from striking such a key is shown in Figure 4. The user will then key in a name of the Font, which may, for example, be up to eight characters.

Next the user will enter detailed specifications for the Named Font, first with respect to the basic CRT which, by way of example only may be the IBM 3270 shown in Figure 1. The basic CRT may not have color or extended graphics. The user will select a normal, bright or unspecified intensity level. The user will move the cursor to select, for example, normal intensity and then press enter.

Likewise, one or more attributes for an extended function CRT, e.g., the color blue, bright intensity, normal highlighting and italics may be selected. The user may select boldface type or a particular character set, if multiple character sets are supported by the particular CRT.

Similarly, data presentation attributes may be selected for the line printer and page printer or for other output devices in the data processing system with which Named Fonts have been associated. By way of example only, for a line printer, density, underlining characteristics and a character set may be specified. It is noted that the characteristic of

"dark" under type density and underlining is achieved by directing the line printer to overstrike data several times on the same line. In regard to underlining, it is noted that "single" refers to a single (as opposed to multiple) overstrike with the underline character.

With respect to the page printer, a type style may be selected and a particular point-size (if available) and boldface may be entered. Italics characteristics may also be selected. In regard to the type style entry, the user may enter a type style designation such as "Times-Roman-Bold" which is converted by the present system into embedded commands which drive the IBM Script system for producing a final form document on a page printer.

It must be emphasized that the Figure 4 display screen for define Font characteristics is only exemplary. Other display layouts could have been chosen for allowing the user to define Font characteristics. In fact, all of the operations of creating and modifying the definition of Formats and Fonts could have been done using directive commands without the benefit of "menu" displays which are illustrated here. In addition, other output devices could have been listed beside those shown in Figure 4. For example, presentation characteristics for an audio device could have been defined, whereby, for example, a high, medium, or low volume characteristic could be specified for a range of words (defining the portion of the text to be vocalized) encompassed by the Named Font.

#### FONT HIERARCHY AND OVERLAYING FONTS

Building on the concept of assigning a Named Font to define the attributes of textual data, the present invention allows the user great flexibility in composing and editing a document by assignment of different levels of Named Fonts and allowing the attributes of different levels to be overlaid and commingled. These levels are arranged so that some of the attributes of the Font at the deepest level may be overlaid by attributes at the next deeper level. As long as a deeper level is not "overlaid" by a contradictory specification, then the deeper level is allowed to "show through", and the various characteristics which survive to the highest level define how the text is actually displayed.

For example, if a Font specifying only "italics" (but which does not name a particular type-style) were superimposed on a Font specifying "Garamond Bold", the result would be "Garamond Bold - Italics", wherever those two particular Fonts were defined to intersect and cover the same portion of text. Then, if another Font specifying "UNDERLINE" (but no other characteristics) were further superimposed, the result would be "Garamond Bold - Italics with Underlining".

However, if a Font specifying the "Helvetica" type-style were further super-imposed over part of the above, that type-style would supersede the "Garamond" (wherever the intersection occurred) to yield "Helvetica - Italics with Underlining".

Fonts can also explicitly specify the absence (or negation) of an option - for example "Underline-Off" (See Figure 4) which would turn off for a specified

range any underlining that might have been activated by a Font at a deeper level. Thus specifying the negation of an attribute has a different effect than simply omitting any specification for that attribute.

As discussed above with respect to Figure 4, a single Font may also specify several characteristics, such as Garamond, Boldface, No underline, Color of blue, Background of Yellow, Flashing, which would supersede similar characteristics at deeper levels (and which could be further overlaid, if desired, by specifications at higher levels).

The method of the present invention allows great flexibility in composing a document. For example, there are at least 14 different Font levels, each of these is specifically assigned a particular meaning (although in the general case, there need be no particular meaning associated with any particular level, and in fact it is within the scope of the present invention to allow a variable, unlimited number of levels of Fonts). It is noted that one of these levels is related to Fonts associated with the Named Formats as discussed above with respect to Figure 3.

The discussion which follows of the fourteen exemplary font levels is for purposes of illustration only. The operability of the present invention is not dependent upon utilizing any one or all of these levels.

Before discussing an example, demonstrating how these Font levels are used, the characteristics of the exemplary fourteen Font levels, starting with the deepest level, are shown below.

#### 1. System Font level.

This Font level is the deepest level and may be overridden by all other levels. The system Font level is not controllable by the user, but rather sets the basic system default Font, if no other Font is specified for a particular portion of the document.

#### 2. User defined Global level.

This Font level may be utilized by the user to override the system default Font. The user may select his own default font which will be utilized if no other font is specified for a particular portion of text.

#### 3. The window font level.

In this Font Level, the word processing editor of the present invention allows the screen to be broken into windows so that different files or different pieces of the same file can be viewed simultaneously. Each window has a particular Font associated with it during this display mode, thereby allowing for easy discrimination during composition (e.g., one might choose to associate fonts defining different colors with different windows).

#### 4. The file-font level.

In this Font level, the user may assign a temporary underlying viewing characteristic to each file (this could be color, type-style, etc.) to allow for easy discrimination between different files during composition. This is useful since different files may be viewed simultaneously or at different times during composition.



##### 5. The base font level.

An indication of this Font level is preserved in the data file itself, and typically provides any default characteristics the user may wish to assign to large sections of the file. This Font level is independent of the Format and Fonts, but can be overridden, in whole or in part, by them. This level will be the level associated with most of the document text and will typically span multiple lines and paragraphs.

##### 6. The Format Font level. (See Figure 3)

This Font level is derived for each line from the underlying Named Format associated with the fields in that line. The Named Format allows each field to potentially be automatically assigned a different characteristic (via the associated named font). Anything defined in the Format level overrides the base level fonts wherever contradicting details occur.

##### 7. The mod Font level.

This Font level is also recorded in the text file, and allows the user to make changes which supersede the format and deeper levels.

##### 8. The footnote Font level.

An indication of this font is recorded in the file. Fonts at this level have a "footnote" attribute, which causes them to be converted to footnotes on hardcopy output devices. In revisable form on CRTs the footnoted material is kept contiguous with the rest of the text. Thus, if the author wishes to visually set off footnotes, he might assign a Named Font with, for example, the attributes of yellow, flashing to any footnoted material.

##### 9. The Index Level.

Textual material controlled by any Font at the "index" level will be compiled into a documents index or table of contents. For example, in a book about swimming, if the term "lifeguard" were to be associated with the index level, it would automatically be placed in the document's Index.

##### 10. The highlight Font level.

This level is similar to the MOD-level except that it supersedes it and can be thought of as being used for short highlights (such as italicized words and highlighted phrases), although there is not actually a strict limitation as to length. It also allows a level of highlighting to be applied to footnotes.

##### 11. The embedded-notes font level.

Text which is annotated with any Named Font at this level is treated as a construction comment private to the author; this material is only displayed when the document is viewed in its (working) revisable form (in final form such as a hardcopy printer or a final form CRT display, the text associated with the font level is omitted). However any such private material is actually recorded in the revisable form text file for later reference. This font level may be given the display attribute of being displayed in a different color so that user can readily find the note.

##### 12. The thinking Font level.

This level is similar to the embedded-notes level, except that the material covered by the font is not stored in the file, and is only designed to temporarily call attention to parts of the document during the current preparation session and relate to notes which are more transient than the (permanent) embedded notes level.

##### 13. The attention Font level.

This level is used to temporarily highlight certain aspects of text -- such as the occurrence of a word for which the user has requested a search, or to set off a misspelled word if the user wants a spell-checking function, etc. The material covered by this font is not stored in the file. This level gives the user the ability to call up positions of the text which need be brought to the user's attention for a particular purpose, e.g., spell-checking, searching, etc. The attention font level will be highlighted to bring the underlying data to the user's attention, e.g., by blinking, underlining, etc. The user specifies the name of the Font (which he has defined) to be used in each of the above described different "attention" situations (e.g., misspelled words, object of a search, etc.).

##### 14. The control Font level.

This is an internal "working" level used to highlight areas which are under definition by the user -- for example while the user is framing (i.e., defining) a range of text to be deleted, or copied, or moved. The text being operated on is highlighted with a particular Font to help the user see exactly what is happening. The user is able to define which particular Named Font is to be displayed at this level for each of the different operations (one for delete, one for move, one for copy, etc.), which may have the attribute of, for example, a special color, or high intensity, reserve-video, etc, associated with each selected Named Font.

#### EXAMPLE OF FONT USE

The sequence of operations for utilizing these different levels is exemplified in Figures 5 and 6. Initially, as shown in flowchart blocks 30, 32 and 34, the user strikes the start paragraph key, selects Format A and types the sentence "Now is the time for all good men to come to the aid of their country". This goes into the text display area of the CRT screen.

By hitting the next paragraph key again, the user may continue the document preparation and select another format, e.g., Format B for embedded quotes. If the user desires initially to produce a document with "good men" italicized, he moves the cursor to "good" (36) and strikes the program function key or keyboard labeled "START FONT" (38).

The system responds with a "WHAT FONT IS THIS?" prompt (40) and temporarily moves the cursor to a specification entry area (41). The user then responds that the Font he desires (which happens to presently define the attribute italics) is

called "SPECIAL". Presuming the user's CRT does not have italicized characters, the font "SPECIAL" may also specify the color "pink" to be displayed for that CRT, whereas the page printers on which the document may ultimately be produced will be controlled to italicize. The user also indicates whether the font is at the BASE, MOD, FOOTNOTE, HIGHLIGHT or NOTES, etc. level (42).

After a Font name has been entered by the user, the system determines whether the font ("SPECIAL") has been defined (44). If the user has not previously defined this Font, he will do so on a screen of the type shown in Figure 4. Upon the user defining the font "SPECIAL" (46), the system automatically enters the control font mode (48) and temporarily controls the CRT display to distinguish the start of textual data which potentially may be operated on by the Font that is being defined, by displaying the text from "good" through the end of the document in a distinct manner (50), e.g., in the color "pink". The system will typically display "pending font range specification" at this point during such an operation (52). Since the system still needs to be informed where the end of the current range of text is to be, the system moves the cursor back to the point in the text ("Good") where it was before the user pressed the "Start Font" key (53).

Next the user moves the cursor to a desired end of font range such as after "men" and strikes the FONT RANGE END program function key (54). It is emphasized that it is within the scope of the present invention to define the range to be associated with a Named Font by commands, by indicating the range start and end with the CRT cursor or by using a "mouse" or any other equivalent technique. The system then displays the text associated with the newly defined Font in a distinguishing manner in accordance with the Font definition for the CRT (58).

It is important to note here that there are no visible, extraneous, control characters or commands inserted into the text. The nature of the material on the author's CRT is implicitly denoted and set apart by (in this case) the color "pink". This could just as easily have been (or could be changed to) set apart by underscore, or reverse video, etc. However, regardless of the visual cues selected by the author during composition, the Named Font "SPECIAL" will result in italicized material when produced on a page (graphics) printer.

If the user desired to further display "their country" in boldface print, the same process would be repeated and a new font would be defined, e.g., "HENRY", where the attribute of boldface print would be selected. The attribute of boldface print would be overlaid on any existing non-conflicting Font for this passage. Of course, since most CRTs do not have boldface characters available, the author may wish to have such material displayed in some other distinguishing color, or perhaps high intensity.

The flowchart of Figure 6 further exemplifies the overlaying of fonts in accordance with the present invention. If the cursor were at the end of the sentence "Now is the time ... country," and the user decided to underline the entire sentence, he could,

for example, hit the "FONT RANGE END" key as shown in flowchart block 70. This would place the system into control font mode (72) and the beginning of the document through the point of the cursor would be displayed in a distinguishing color or manner (74).

The user will next move the cursor to the beginning of the sentence (76) and strike the "START RANGE FONT?" key (78). The system will recognize that a font has yet to be selected, will display "WHAT FONT IS THIS?" (80) and will move the cursor to the specification entry area of the screen (81), where the user, desiring to underscore the sentence may, for example, type "US". The user may also select the highlight font level by entering a symbol such as "H" (82). The user then will define a FONT "US" designating the attribute of underscoring (86) or call such an underscoring Named Font from the system library of fonts. It is noted that, in the usual case, the Font is already defined so that the user does not need to designate its characteristics except, at most, the first time it is referenced. The system will then return the cursor from the specification entry area to its previous position in its text.

Presuming that the font "US" only specified the attribute of underscoring, our sentence "Now is ....country." would be underscored and the previously discussed attributes of "good men" being italicized and "their country" being in boldface would additionally percolate up and remain in effect (88) when the document is produced on a laser-type printer. On the CRT, of course, these Fonts would likely be represented by special colors and display intensities.

The user has thereby specified three user levels of Fonts with respect to the same document portion. Although the number of levels so specifiable may be unlimited, the present invention contemplates the levels listed above which are specifiable by the user for overlapping textual material to be limited to a readily manageable level such as six to ten levels. If two Fonts are specified at the same Font level for the same range, then the most recent specification will override or reset the previous specification (but only to the extent the ranges overlap).

Figure 7 is a block diagram of the software subsystems which control the word processing document composition editor of the present invention. In the description which follows in order not to obscure the present invention, word processing control function details which are conventional and which do not assist in the understanding of the operation of the present invention will not be described in detail. Such conventional details will be readily understood and may readily be implemented by those of ordinary skill in the art. The subsystems shown in Figure 7 may either be permanently stored in main storage, or may be loaded into main storage from external storage prior to the execution of their respective routines.

The components in Figure 7 will be described generally first and then in more detail with respect to the figures which follow. Turning first to file manager 100, this subsystem keeps track of the location and

reading and writing of data records in the file. The file manager is responsible for the efficient insertion and deletion of records. When records are inserted at a particular portion of the file, the remaining records must be shifted, either physically or logically, to accommodate the new records. The file manager keeps track of external or permanent "points". "Points" associated with a Font denote the document point where the Font begins and the document point where the Font ends. The "permanent" points refer to places in the document where a font has been set by the user and which involves a Font level that is permanently retained in the file. Fonts which have been set at the Base Font level the Mod Font, the Highlight Font, the Index Font, for example, are permanently retained in the file. The start and end range "points" of, for example, a Base level Font are monitored by the file manager.

The file manager 100 keeps track of records by way of a pointer or address which specifies where each record is stored. Each record has the structure as shown in Figure 8. The records are configured such that one record portion 102 stores the textual data and another record portion stores control information 104 relating to the "points" associated with the record. The text data is stored essentially in the same manner as it appears on the CRT screen, although this does not necessarily need to be the case.

The control information includes Format identifier 106, which defines the Format associated with the text. The identifier may include an index field, which points to the memory address where the format definition control statement is stored. Additionally, the control information includes point control data fields (108, 110 and 112) for the points associated with the record.

The record control information for any particular point defines the offset 114 (i.e., location relative to the first byte of the record) of the point in the text, the type of the point 116 (e.g. start of Font range, end of Font range), the level of the Font (base level, highlight level, footnote level, etc) and the identifier of the Named Font 118.

In the exemplary implementation, in addition to points which may be defined for a particular record, the control information also contains the names of whatever Fonts may be outstanding i.e., active, at each of the recorded levels (base, mod, highlighted, footnote, index, notes, etc) at the close of the record. This snapshot indicates the status of each Font level 120, whether it is off or on, and if on, what the Font ID is.

It is noted that it is within the scope of the present invention for the records to be structured so that the point defining information is actually embedded in the text as opposed to being stored as offsets from the start of the record or file. The essential aspect is that place marks, although associated with the text, and which define the nature and the range of Fonts associated with the record data, are not displayed to the user during composition.

The point manager 200 in Figure 7, like the file manager 100, keeps track of points in the file associated with Fonts at predetermined levels. The

Font levels for which the point manager is responsible are the control Font, the attention Font, the thinking Font and the system Font levels.. The point manager 200 keeps track of points in the file but the associated Fonts are not stored so as to be permanently associated with the document file, but rather are stored in an auxiliary storage. These Fonts may be thought of as being too transient to be permanently associated with the document file, e.g., the attention and control font levels relate only to the display during the current preparation session.

The point manager 200 keeps track of these more transient fonts with the use of point manager elements which contain much the same information as the offset and type of point information handled by the file manager. As will be explained further in regard to Figure 11, these point manager elements are stored in a chained or linked fashion in which one point managing element points to the next point managing element in the chain.

As shown generally in Figure 9, a point managing element PME includes at least four fields and defines the file I.D. 202, which identifies the document being prepared, the record I.D. 204, and offset 206 which identifies a record within the file, and the offset within the record, i.e., the byte where the point starts or ends, the type of point information 208 and the identifying the name of the Font (or a pointer to where the Font specification is stored in auxiliary storage). The type of point information may include a wide range of control information concerning the point (e.g., see 110-118 in Figure 8).

The point manager 200 also keeps track of single points (as opposed to points defining a range). In this fashion, the user can define a "bookmark" point by a designated name and then go back to that point in the text by calling for the designated name. The bookmark label is also stored in the PME.

The file manager 100 and the point manager 200 must interact when the file manager adds or deletes records. As will be readily apparent to those skilled in the art, the insertion or deletion of text and/or records by the file manager 100 may require the point manager 300 to shift the record I.D.'s in a coordinated fashion with the file manager so that the point manager accurately keeps track of text points (offsets and record IDs) as they shift positions in the file.

The system shown in Figure 7 also includes a display and output manager 300. The display and output manager 300 keeps track of data which is to be displayed on the screen. In addition, this subsystem uses the parallel definitions of Formats and Fonts for the systems other output device and controls their respective output.

With respect to the CRT being used for document composition, the display manager 300 is responsible for insuring that the text is displayed in the manner in which it should be shown (e.g., a distinctive display, such as change of color or a flashing attribute in the control font mode). The display manager also responds to user commands which call for modifying the display of data (e.g., insert, delete, move, copy etc.).

As will be explained in further detail below, the

display manager 300 must read the records, and build a font vector defining all the currently active fonts that exist for a given record at each font level (e.g., Default, Base, Format, Highlight, Mod, Footnote, Notes, Attention, Control etc.), beginning at the point where the record starts. The display manager obtains this information from the file manager 100 and/or the point manager 200. The initial font vector is built for each line which is being displayed. As the line is laid out on the screen, the active fonts relating to the next line to be displayed must be merged therewith.

The font vectors may be stored as a list in memory and these font vectors are loaded with information from the point manager 200 and the file manager 100. The point and file managers must be accessed to determine what Fonts were active for the records which are being displayed. Thus, the display manager 300 builds a line on the screen by determining the Fonts at each level which are active at the beginning of the record (by looking, e.g., at the Fonts active at the end of the previous record). Next, the display manager controls the display of each byte of the record and merges the information regarding the starting and stopping of the fonts from the point manager 200, the file manager 100 together with the format information (which defines the beginning and ending of the Format field and is contained in the stored Format specification). By such correlation, the final display for each character of each line is created.

The input and command manager 400 shown in Figure 6 must read the input data from the CRT keyboard and decode the commands from the program function keys or commands which are otherwise entered. The input manager 400 determines whether an entry will have global effects, i.e., will require a reformatting of other records due to, for example, inserted or copied data. The input manager 400 interacts with the file manager 100 and point manager 200 of such instances so that the stored point information may be correspondingly adjusted. If necessary, the input and command manager 400 will reformat data in lines subsequent to the current lines into which data has been inserted according to each format associated with such subsequent lines until the end of a paragraph is reached. Such shifting requires that the points in the point and file managers be likewise adjusted.

#### FILE MANAGER

As noted above, the file manager keeps track of data records in the file. The file manager 100 as shown in Figures 10 A-C, serves to respond to calls (120) originating from the display manager 300 or the input manager 400 to retrieve the record currently being operated on based upon a specified record I.D. The file manager returns the record to the address indicated by the record I.D. after processing (122). Likewise, the file manager 100 also fetches when requested (124, 128) the successor to the record with the specified I.D.(126) as well as the predecessor to the record with the specified I.D.(130).

As shown in Figure 10D, if an insertion command

has been executed (140), such that a new record needs to be created, the new record and new record I.D. is inserted into the file after the specified record( 142, 144). Additionally, the file manager adjusts all point manager block elements shown in Figure 9 to reflect any modified address information due to the insertion. Points are also subject to adjustment even when a small amount of text is inserted which doesn't actually create additional records.

The file manager similarly responds to any decoded deletion command (108) by deleting a range of text from one or more records and by updating all point manager elements for text and records which were shifted due to the delete operation (150, 152). It is noted that once a record is deleted, the point managing elements associated with that record are likewise deleted. It should be noted that moving and copying text is a combination of the delete and insert function.

#### POINT MANAGER

Turning to further details of the point manager 200, as noted above, this component keeps track of points which are associated with Font levels and other miscellaneous points of interest which are not stored with the file records. The point manager 200 stores point blocks 220, 234, 236 which, as shown in Figure 11, are chained together.

Each point block (e.g., 220) includes a chain address 222 which links one block to at least one other block. The block 220 further includes a pointer 224 which identifies the location in storage of the document file control block, a record identifying address 226, and a record offset 228 which identifies the point, i.e., the byte relative to the beginning of the record at which the point is located. Flag location 230 indicates the type of point (such as beginning or end indicators). Location 232 identifies the name of the font and the Font level. With respect to block 234, this linked block may define the end of a Named Font range.

Where multiple fonts are involved, a particular point must be linked with a number of different blocks rather than just a single block. Such linked points may be located within a single record or may span records.

The point manager 200 may be called upon by the input and command manager 400 to create a point in response to the depression of a "Create Point" program function key. Thus, when a Create Point key is depressed, a point block element will be created for the document point at which the CRT cursor is situated. This point which may denote the beginning of a Font Range, will be a temporary point while it is in the Control Font Level (see the definition above), i.e., while the user is in the process of specifying what is to be done with the range being defined. Likewise, a point block element is created for the point denoting the ending of the range. After a Font Name is associated with such a range, the Font is fully defined.

It should be remembered that the file manager 100 likewise keeps track of points. The points associated with the file manager, however, are "permanently" associated with the file. Like the point data stored

with the file, the point manager 200 also stores Font level information (e.g., control, attention, etc.).

As shown in Figure 12, the point manager 200 uses file I.D., record I.D., offset within the record and type of point information as the input parameters for creating a point (240). The type of point information includes the range start, range end, Font I.D. and Font level indicia. The point manager 200 allocates a block of storage for the block and loads the above information into the block (242). The block is then positioned in the correct order. As shown in 244, the point manager positions and chains the allocated block to related blocks based on the file I.D., record I.D., offset and the point type information. During the positioning processing, if one point denoting a Font ending occurs at the same spot as another point denoting the beginning of a Font, the end point is recognized before the start point, and is assigned a point block element in the chain ahead of the point denoting the Font beginning.

When a point is deleted, the point manager removes the deleted point from the chain and frees the previously allocated storage. The point manager will typically be called upon to create a point by the input and command manager 400.

#### DISPLAY MANAGER

The display and output manager 300 controls the display of information according to the points which are provided as input. The first aspect of the display manager control is the movement of information to the proper display area as the document is constructed. The display manager positions the point at a desired position on the display window and builds the display by acquiring records from the file manager. The display manager expands the acquired records and consolidates the Fonts.

The display and output manager 300 additionally includes a subsystem which translates the expanded record format into a format having embedded command codes such that the aforementioned IBM Script Document Composition Facility (DCF) can take such output records and produce a hard copy on auxiliary devices, such as, for example, the IBM 4250 Electro Erosion page printer 8 shown in Figure 1. Thus, to produce a hard copy document, the display and output manager 300 automatically takes the definitions of all the Named Formats and Named Fonts and creates a file having embedded commands suitable for driving the IBM Script DCF, a process which heretofore was highly labor intensive, having to be done manually.

To consolidate the Fonts and produce a CRT display, initially the display manager acquires a display window alignment specification to determine from what point on the display screen the display should commence, e.g., a display line near the top of the screen. The point manager element corresponding to the point in the record to be displayed is then acquired. This element specifies the record I.D. and the offset information which is used to determine the desired position of the display. This point identifies the record as, for example, being record 100 and may reference a particular word in the record. The referenced word is then taken and displayed on an

anchor line on the display, i.e., a user chosen preferred location on the display to begin display on the display window as indicated by the aforementioned display window alignment specification. The display manager then accesses the file manager 100 to retrieve the previous record and the font specification associated with the previous record, goes through all the point manager elements to similarly find all the font specifications which are in effect at the start of the current record.

In this process, the display manager 300 finds the font specifications at each Font level which are logically open at the beginning of the record, i.e., where a Font specification start range has been set in a previous record but has not yet been closed. Then the display manager does an analysis to merge the Font information from the following three sources: a) the record information itself, b) the point manager element information, and c) the format information as defined by the stored format specification.

The display manager sorts this information based on the respective offsets within the record. Next the display manager goes through the record to be displayed on a character by character basis and determines if any change need be made at a particular offset within the record. If the offset information indicates that a change needs to be made to the character in question, then such a change is made based upon the font specification identified by the point. The display manager then starts at the deepest Font level and superimposes or overlays each of the display attributes defined at each level on the characteristics of the aggregate of the preceding levels. In a similar fashion, the display and output manager performs a corresponding output managing process to control the distinctively defined display or hard copy characteristics specified by the Named Formats and Fonts for each of the output devices in the system.

This display and output managing processing performed by the display and output manager 300 is detailed in the flowcharts of Figures 13-15. As shown in Figure 13, the display manager determines the record to display on the first line of the display (310). The first line on the screen on which data may be displayed will be preceded by display lines allocated to a command line, an error line, a header line which indicates the file being operated on and a Format line which displays the active Named Format associated with the data record occupying the window's anchor line. The display manager calls the file manager to retrieve the desired record and the previous record (312) and checks to determine the last point ranges in effect prior to the start of the record (313).

Next as shown in Figure 13, the record is expanded and all Font specifications associated with the record are merged (314). This process is further detailed in the flowchart of Figure 14. An exemplary routine which accomplishes the function of expanding the record and merging all associated font specifications is set forth in Appendix "A" as the MAPOLINE routine. Next the display manager checks to determine whether the display window is now filled (316). If so, the display manager exits the

routine. If not, the display manager calls the file manager 100 to retrieve the next record (318). The display manager next checks to determine whether the end of the file has been reached (320). If not, it loops back to expand the next record. The display manager exits the routine when the window is full, or the end of the file is reached.

One manner in which records may be expanded by the output display manager is shown in further detail in Figure 14. It should be noted that the records are expanded so that they may be placed in a format (e.g., as required by the IBM script DCF system) for producing a final form hard copy or display for each output device being relied on for document production. Thus, the display and output manager 300 will reformat record information based on the previously defined Format and Font specifications for each target output device as for example shown with respect to the Named Format specification of Figure 3 and the Named Font specification of Figure 4.

As shown in Figure 14, to expand the record, initially the Format must be obtained based upon a format identifier associated with the record as shown in Figure 8. Next the display and output manager determines the fonts which are in effect, (i.e., a font range which has been started but not yet ended) at each Font level at the beginning of the record (336). The display and output manager makes this determination based on information obtained by scanning point manager elements associated with previous records obtained from the file manager (332, 334).

The display and output manager, in scanning the point manager elements and the records to find all fonts which have been opened and not closed prior to processing the current record (336), compares the record I.D. of the current record being processed to find a less than or equal record I.D.

Next, as shown in Figure 14, the display and output manager creates a font vector, i.e., a hierarchical stack of fonts in the order of priority based on the previously discussed Font level hierarchy (338). As previously noted, the lowest priority font level is the system default level and the highest priority level is the control Font level. The GATHFONT listing in Appendix A shows an illustrative routine for gathering Fonts (e.g., the routine gathers attention and control Fonts) that are active to perform the functions shown in blocks 332-338. The GATHFONT routine is called by the aforementioned MAPOLINE routine to perform such functions.

The display and output manager 300 will, after initializing the Font vector stack, loop through the stack vector from the lowest to the highest level to determine whether a Font has been defined at the particular Font levels. If a Font has been defined, then all specified attributes are merged into the vector stack in accordance with a merging of attributes routine (340). An exemplary routine which accomplishes the function of merging Font attributes is set forth in Appendix "A" as the MERGFONT routine.

As shown in Figure 15, the merge attributes

routine is called and record processing is initiated (350). As each byte of the record is scanned (352), a check must be made to see if that position in the record is associated with a change to any font.

This determination is made based on a three way collation of point manager elements, points which are defined with the record and the stored format associated with the record (354). The point manager elements are retrieved from the point manager blocks which are chained together based on record I.D.'s and the offsets within the record which defines the points locations. The points within the record are obtained by accessing the control information associated with each record as shown in Figure 8.

The display and output manager merges this information together. As indicated in Figure 15, the display and output manager determines whether there is a defined change to any Font, e.g., an indication that a Font range must end (356). If not, then as shown in Figure 15, the byte is displayed (or printed depending on the device in which the document is being produced) with the existing composite font (360), e.g., one which defines attributes for the byte to be high intensity and underlined. If there is a change in the Font, then the change is reflected by changing the appropriate slot in the Font vector. Then the merge attribute routine must be called again to recompile all entries in the Font vectors and then the byte will be displayed using the revised composite Font 358. After the data is displayed, the display manager will look at the next byte until the end of the record or until the end of the display is reached.

## INPUT MANAGER

As shown in Figure 16, the input and command manager 400 operates on an input data stream after the user has pressed "enter" or used a program function key (410). The input manager merges the input data into the existing text to reflect changes, insertions and deletions. As necessary, the point manager elements and the record pointer elements are adjusted to reflect the insertion or deletion of previous bytes (414).

An entered command string is saved by the input handler and decoded (416 and 418). Next the input manager (400) determines whether the input should be mapped back to the record before executing the command (420). If the answer is no, the command is executed immediately (422). Although the majority of commands will require a remapping of the data, certain commands such as a "TAB" command are executed immediately, it being unnecessary and undesirable to rearrange or rejustify text before the cursor is moved to the tab position.

Entering further data into the record, may require a certain field to be rearranged. Under such circumstances, as shown in Figure 17, the input manager splits fields as necessary based on the Format currently in effect for the line (424). Thus, the rules established by the Format in effect for the line are applied to the new characters added to the line.

As shown in Figure 17, the input manager must adjust all pointers to reflect the movement of data due to the newly entered or deleted data and to the



associated splitting or merging or regrouping of fields and records. In order to accomplish this, the input manager adjusts the point manager element pointers for the points which are stored external to the file. Likewise, the points defined in the file are changed to reflect the shifted or moved data. The input manager thus merges input data to reflect insertion and deletion of text. This may involve creating or deleting records and/or rearranging text (426). The data is also aligned, expanded, if necessary to fit the appropriate field, hyphenated, checked and verified according to the Format field rules discussed above (428) (such checking and verifying may, for example, include spell checking).

The input manager then executes any command initiated by the depression of a program function key or a command keyed in and entered by the user (430). The execution of any such command may, of course, also result in data insertion, deletion, rearrangement, etc. The functions initiated by the execution of a command relating to the Named Fonts and Formats have been explained in conjunction with the preceding description of the present invention. Some other commands shown below relate to conventionally performed functions in word processing systems (e.g., scroll commands) and will be apparent to those skilled in the art. The commands typically executed by the input and command manager 400 include, but are not limited to, the following:

- Start Range for Font (at a particular level)
- End Range for a Font Specification
- Start Range for Delete
- Start Range for Move
- End Range for Move
- Start Range for Copy
- End Range for Copy
- Start Range for Spellcheck
- End Range for Spellcheck
- Next Paragraph
- Scroll Forward
- Scroll Backward
- Scroll Left
- Scroll Right
- Jump to a Line
- Jump to Top
- Jump to Bottom
- Jump to a Point Previously Defined
- Define Format
- Define Font
- Change Format Associated with a Line
- Change Format Associated with all Lines in a Paragraph
- Change Format Associated with all Lines in a Range
- Change Font Association
- Display All Named Formats
- Display All Named Font Definitions
- Display Detailed Specification of Format
- Update Detailed Specification of Format
- Change the Command Associated With Program Function Key
- Display the Program Function Key Command
- Highlight Level Select
- Base Level Select

- Mod Level Select
- Note Level Select
- Create Point

Based on the above detailed description of the invention including the flowcharts relating to the file manager 100, point manager 200, display and output manager 300, and the input and command manager 400, those of ordinary skill in the art will appreciate that specific programs may be routinely written for a given data processing system to implement that which is claimed. By way of example only, the assembly language routines designed to run on an IBM 370 which have been referred to above in the specification are shown in Appendix A below. Data structures which are utilized by the routines in Appendix A are shown in Appendix B.

#### APPENDIX A

The exemplary MAPOLINE, GATHFONT and MERGFONT routines referred to above are set forth below:

#### APPENDIX B

The above routines utilize the following data structures: POINTDEF (defines the various types of points that can exist), PMR (position marker element which defines the point manager elements), FASE (Font attribute specification element) and FMTSPEC (Format specification). These data structures are shown below.

While the present invention has been described in terms of its presently preferred form, it is not intended that the invention be limited only by the described embodiment. It will be apparent to those skilled in the art that many modifications may be made which nevertheless lie within the spirit and intended scope of the invention as defined in the claims which follow.

#### Claims

1. In a data processing system for composing documents having multiple lines of alphanumeric data, means for entering data and commands relating to a document to be composed (2, 4), data processor means for processing said data and executing commands (10), memory means coupled to said data processor means for storing data and commands (10-main memory portion, 16, 18), and at least one output means for presenting the document (2, 4, 6, 8, 12, 20), said at least one output means including a display device (2, 4), said data processing system being characterized by:

a. means for assigning a first name to represent and be associated with a first set of data presentation characteristics (2, 4, Fig. 2, Fig. 4);

b. means for selecting a first set of data presentation characteristics to be associated with said first name (2, 4, Figs. 2-4);

c. means for displaying on the display

device a portion of the text (10, 300);

d, means for defining the beginning and ending of at least one range (2, 4, 10, 100, 200) delineating the document portions for which the first set of characteristics are desired by associating place marks with the document text which are not displayed with the text to the document composer during composition and which define at least the range beginning and the range ending; and

e. means responsive to the entry of said first name for interactively presenting the data in the defined range with said first set of characteristics (10, 300) on the composer's display device (2, 4); and

f. means for insuring that all other document portions previously associated with said first name are presented having said first set of data presentation characteristics (10, 100, 200, 300).

2. Apparatus according to claim 1, wherein said composer's display device is a CRT (2) and wherein said means for selecting a first set of data presentation characteristics includes means for displaying on the CRT a detailed specification for said first name (2, Figs. 2-4), said detailed specification containing a plurality of characteristic selection areas (Fig. 4); and

means for selecting data presentation characteristics (2, Fig. 3) for at least a plurality of output means (2, 4, 6, 8, 12, 20) including said CRT (2) to thereby associate data presentation characteristics with said first name.

3. Apparatus according to claim 2, further including an additional CRT (4), a line printer (6), a page printer (8) and an audio device (20) and means for selecting data presentation characteristics for at least one of said additional CRT(4), said line printer(6), said page printer(8) and said audio device (20).

4. Apparatus according to claim 3, further including means for selecting (2, Fig. 4) at least one of type density, type style, type size and underlining features for the line printer(6) or page printer (8).

5. Apparatus according to claim 1, further including means for associating a first predetermined level (42, footnote or index levels) with said first name and means for placing textual data (300) associated with said first name in at least one of a footnote format or in an index when said document is presented on at least one of said output means.

6. Apparatus according to claim 1, further including means for assigning a hierarchical level to the assigned first name (42, 300).

7. Apparatus according to claim 6, further including

means for assigning a plurality of additional names to be associated with a respective plurality of sets of data presentation characteristics (2, 4, Fig. 4);

means for associating at least one of said plurality of additional names with at least a

portion of said range (100, 200);

means for assigning a hierarchical level to said at least one of said additional names (2, 4, 42) which is higher than the level assigned to the first name;

means for presenting both the characteristics associated with said first name and said at least one additional name as long as the characteristics of the first name are not overlaid by a contradictory specifications of the said at least one additional name (350); and

means for superseding the characteristics associated with the lower level by the characteristics associated with the higher level if the characteristics are contradictory (356, 358, 360).

8. Apparatus according to claim 1, wherein the document is stored in the memory means as a plurality of records (102, 104) and further including means for associating control information (108, 110, 112) with each record relating to the points in which a name associated with a set of data presentation characteristics begins or ends, said control information including an offset (114) defining the bytes of the record relative to the first record byte that the characteristics begin and end, and an identifier that identifies said name (118).

9. Apparatus according to claim 8, including means for storing with each record an indication of the names which are active at the end of each record (120).

10. Apparatus according to claim 1, further including manager means (100, 200) for keeping track of all points in the record which relate to a range for an assigned name representative of a set of data presentation characteristics.

11. Apparatus according to claim 10 further including a file manager means (100) for keeping track of points in the record relating to names assigned to any of a first group of hierarchical levels and a point manager means (200) for keeping track of points in the record relating to names assigned to any of a second group of hierarchical levels.

12. Apparatus according to claim 10, including means for storing control information relating to each of the points and linking related points together (220).

13. Apparatus according to claim 1, including means for storing all the assigned names, their associated data presentation characteristics, and the document portion encompassed by the names (10-main memory portion);

display manager means (300) for merging all data presentation characteristics relating to a predetermined portion of the document: said display manager means further including means for presenting the data in accordance with a composite set of characteristics (350).

14. Apparatus according to claim 1, further including:

means for modifying (2, 4, Fig. 4) said first set of data presentation characteristics associated with said first name and means for



responding to the entry of the modified characteristics to automatically present all document portions associated with said first name to reflect said modified characteristics (100, 200, 300, 400), whereby all data associated with a given name is presented so as to reflect said modified characteristics after changing only a single name specification and wherein such changes are displayed interactively in real-time during composition provided such changes relate to the composer's display device characteristics.

5

10

15

20

25

30

35

40

45

50

55

60

65

15

0278722

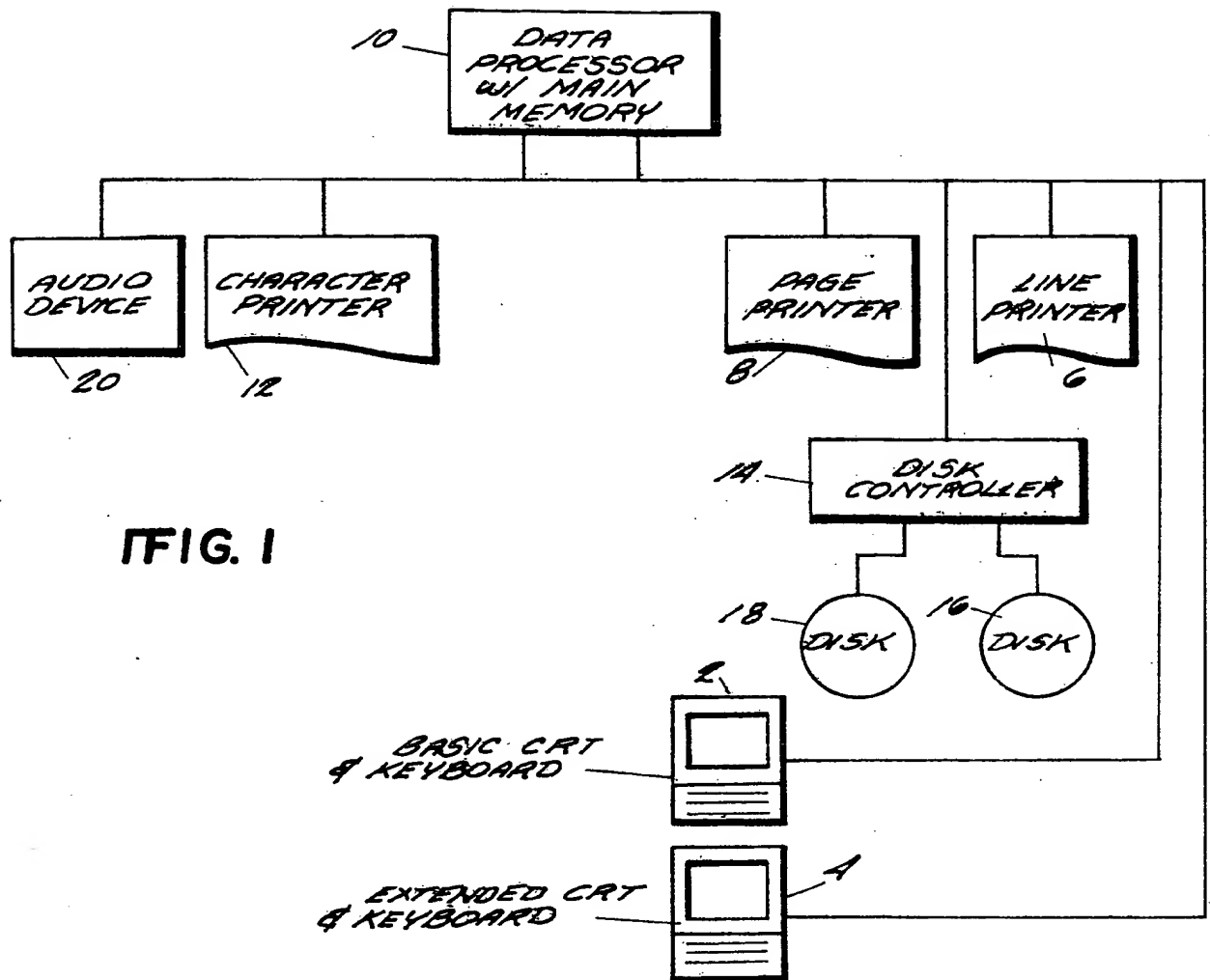


FIG. 1

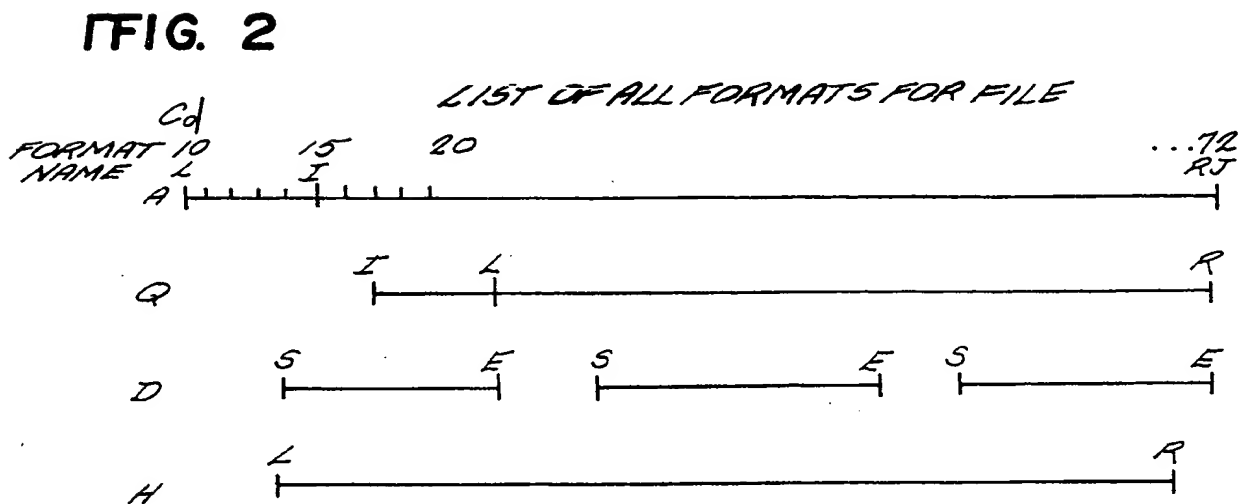


FIG. 2

0278722

**FIG. 3**

## DETAIL SPECIFICATIONS FOR FORMAT A

FIELD	FONT ID	CHAR.	BASIC CRT	EXT CRT	LINE PRINTER	PAGE PRINTER
1	GEORGE	L,R,H,W	1 TO 69	1 TO 120	1 TO 132	.5 IN TO 8.0 IN
		SP.				
2			TO	TO	TO	TO
			TO	TO	TO	TO
			TO	TO	TO	TO
			TO	TO	TO	TO
			TO	TO	TO	TO

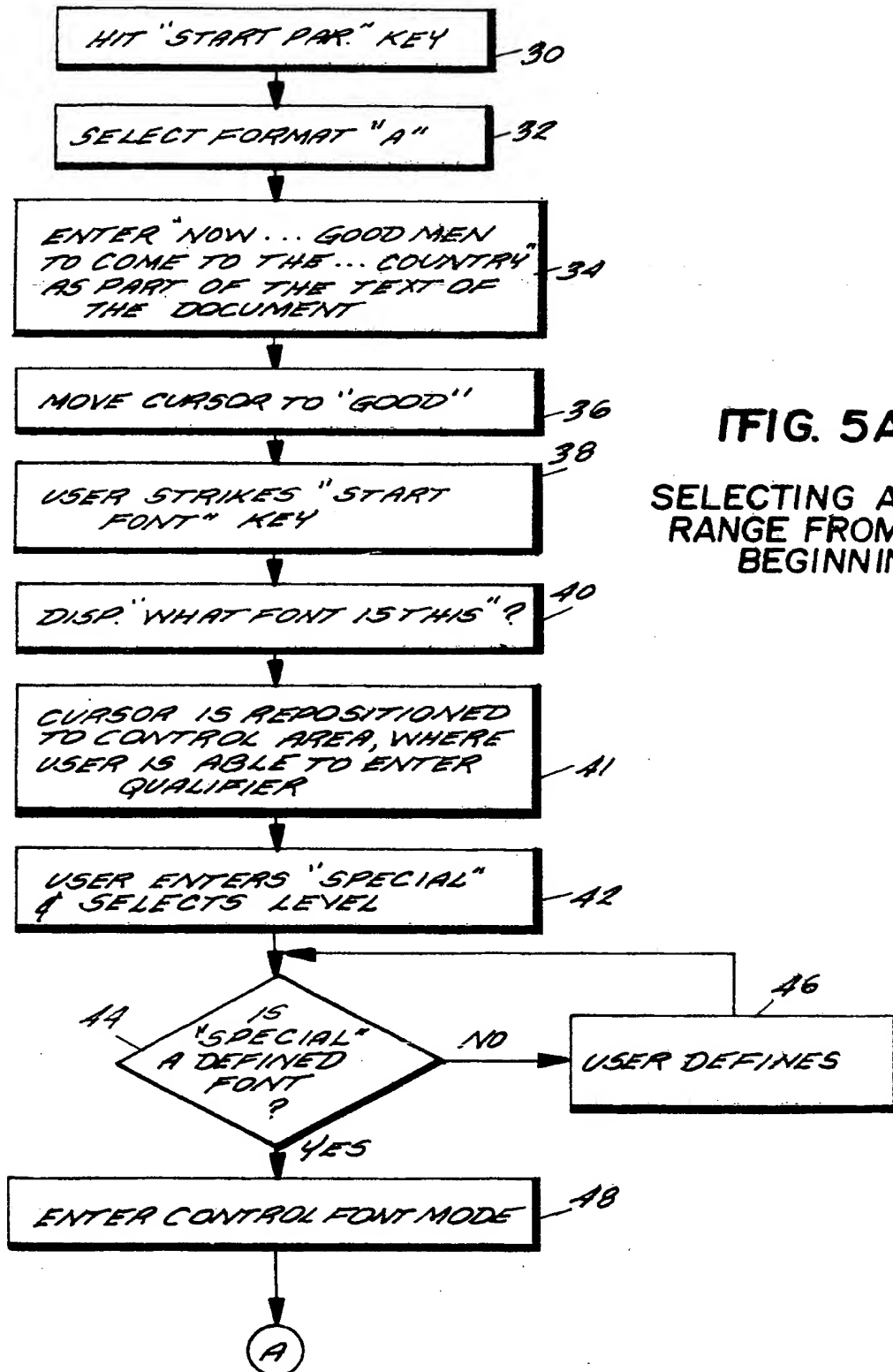
0278722

# DETAILED SPECIFICATIONS FOR FONT ID = BEN

BASIC CRT	EXTENDED CRT	LINE PRT	PAGE PRT
<b>INTENSITY</b> -- UNSPEC -- NORMAL -- BRIGHT	<b>COLOR</b> -- UNSPEC -- NORMAL -- BLUE -- RED -- PINK -- GREEN -- TURQ -- YELLOW -- WHITE  <b>CHAR-SET:</b> _____ ITALICS: -- UNSPEC -- YES -- OFF  <b>INTENSITY:</b> -- UNSPEC -- NORMAL -- BRIGHT  <b>HIGHLIGHT:</b> -- UNSPEC -- NORMAL -- BLINK -- REVVID -- UNDLIN	<b>DENSITY:</b> -- UNSPEC -- NORMAL -- DARK  <b>UNDERLINE:</b> -- UNSPEC -- OFF -- SINGLE -- DARK -- =TEXT  <b>CHAR-SET</b> _____	<b>TYPE STYLE:</b> _____  <b>POINT-SIZE:</b> _____  <b>ITALICS:</b> -- UNSPEC -- YES -- OFF  <b>BOLDFACE:</b> -- UNSPEC -- YES -- OFF

FIG. 4

0278722



**FIG. 5A**  
SELECTING A FONT RANGE FROM THE BEGINNING

0278722

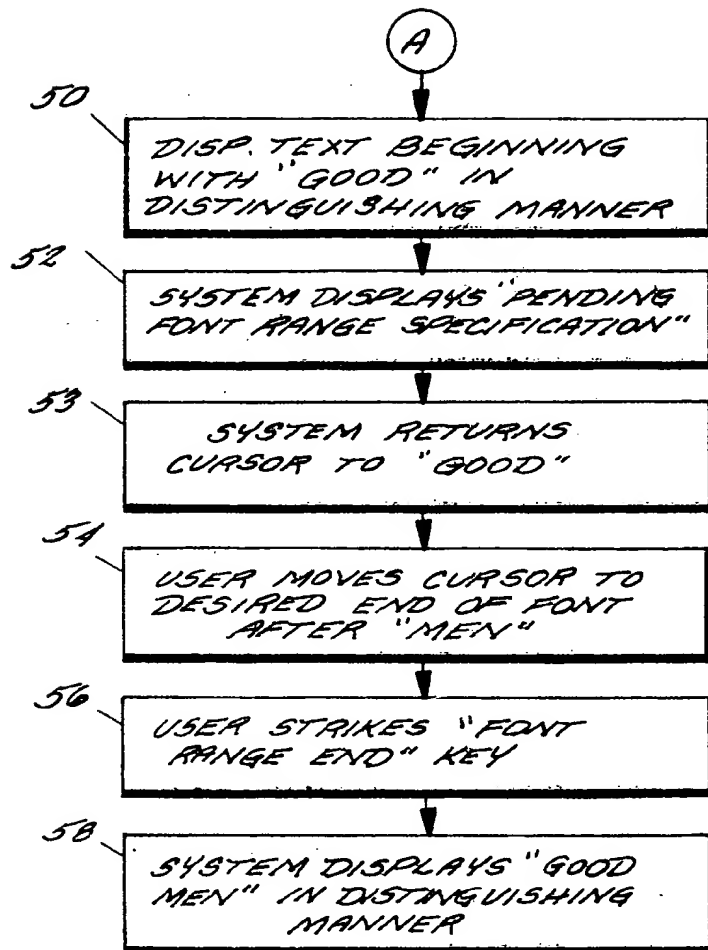


FIG. 5B

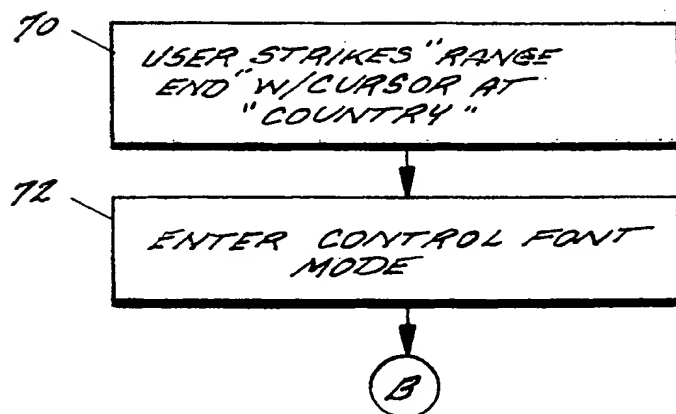


FIG. 6A

SPECIFYING A FONT RANGE STARTING WITH THE END OF THE RANGE

0278722

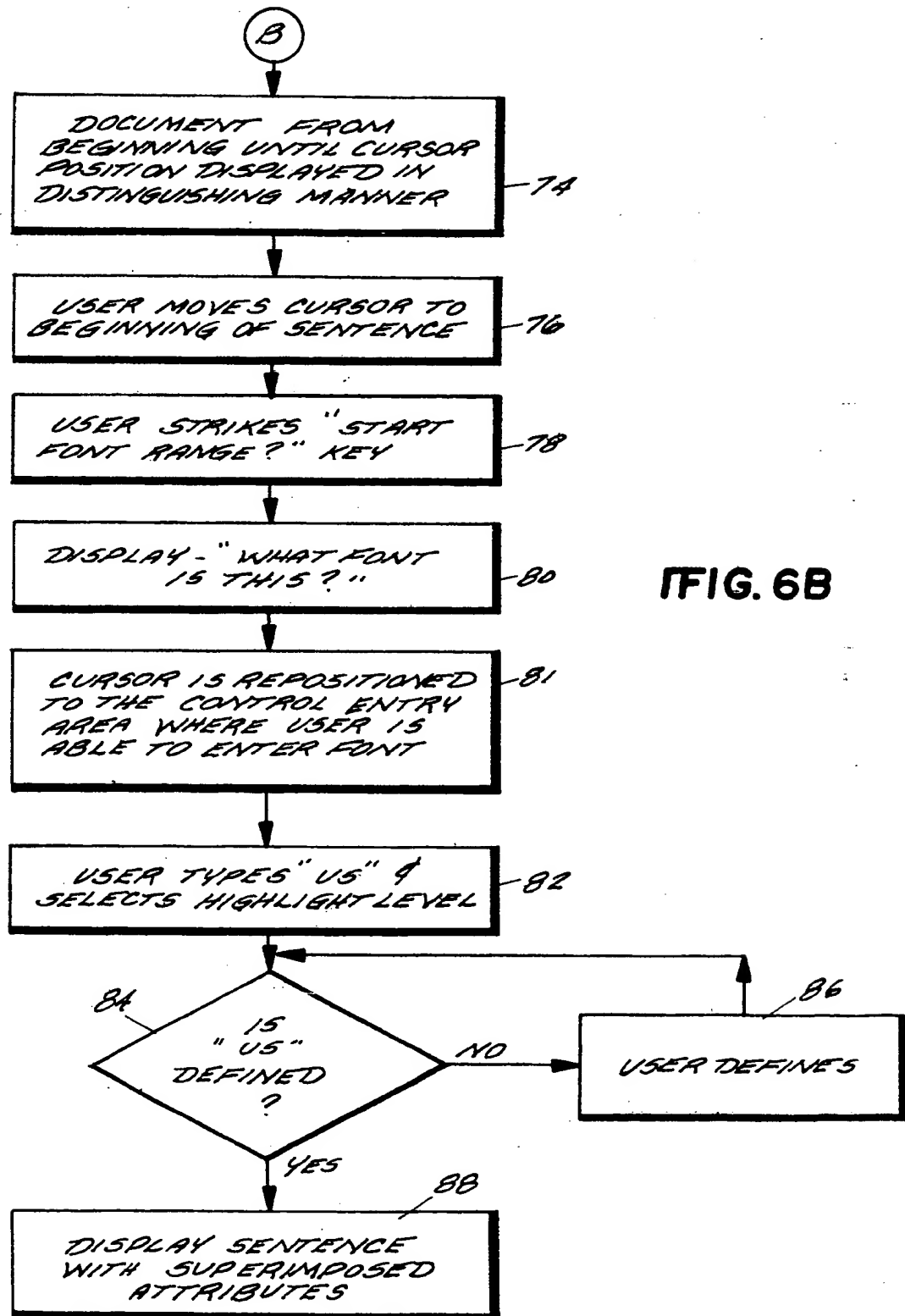


FIG. 6B

0278722

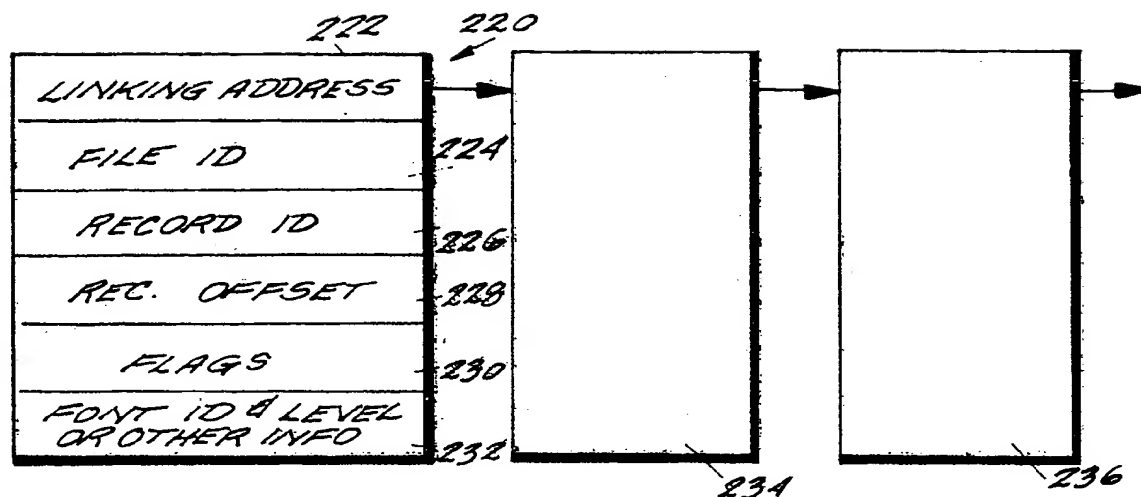


FIG. II

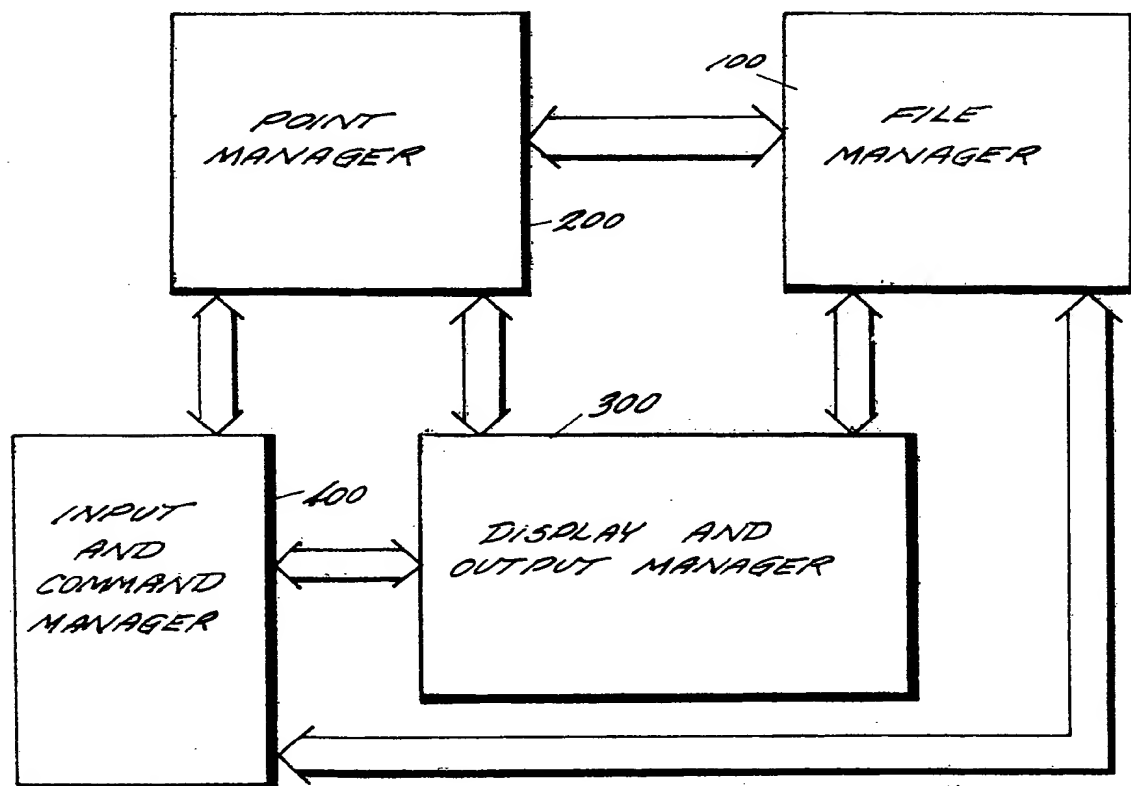
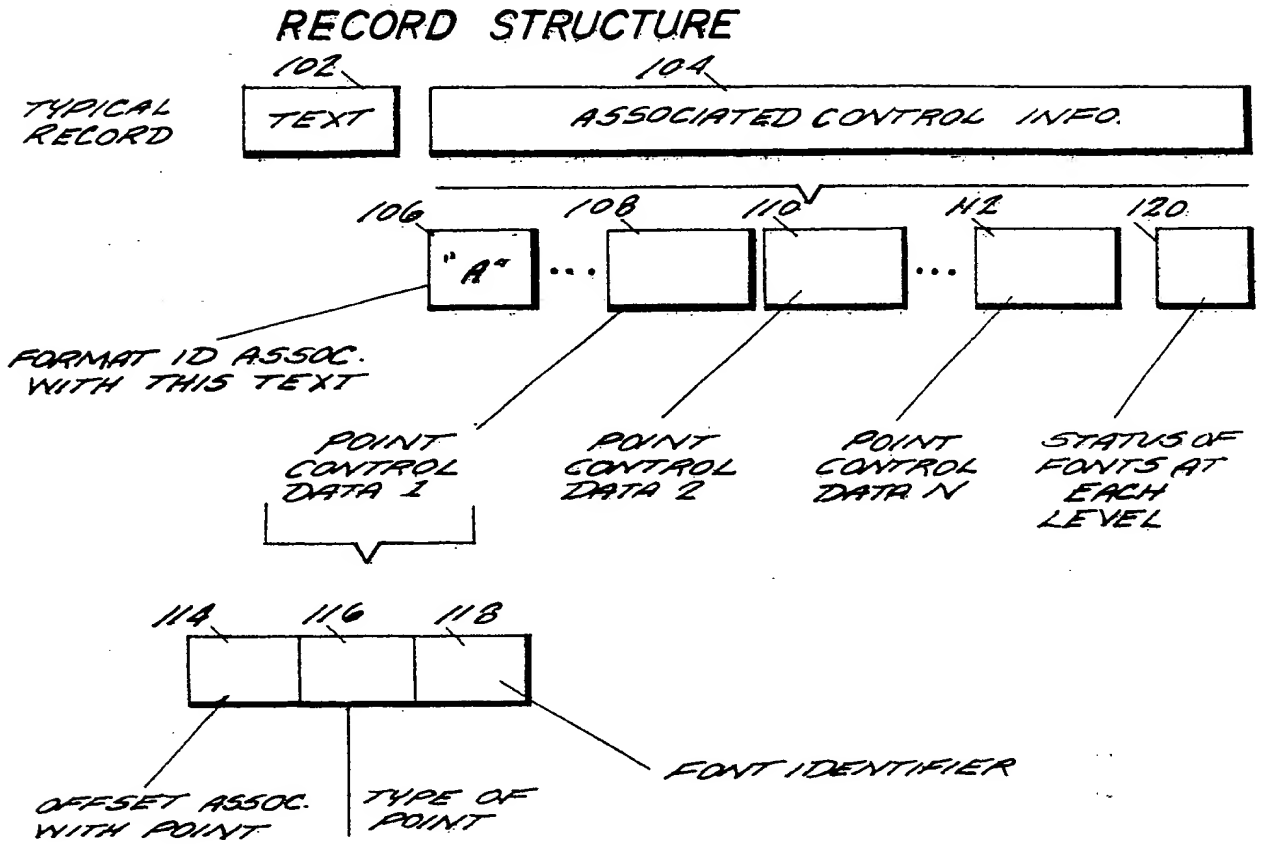


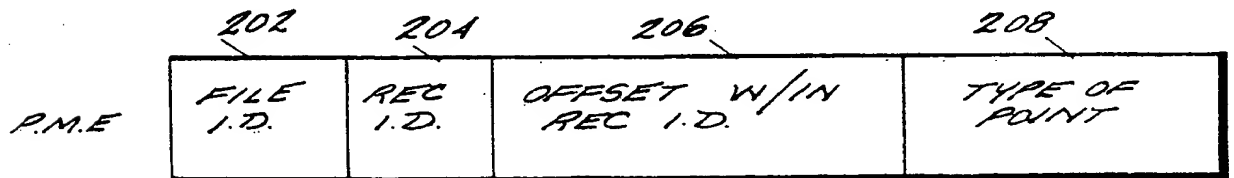
FIG. 7



0278722



**FIG. 8**



**FIG. 9**

0278722

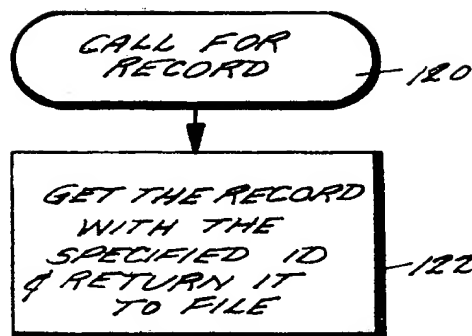


FIG. 10A

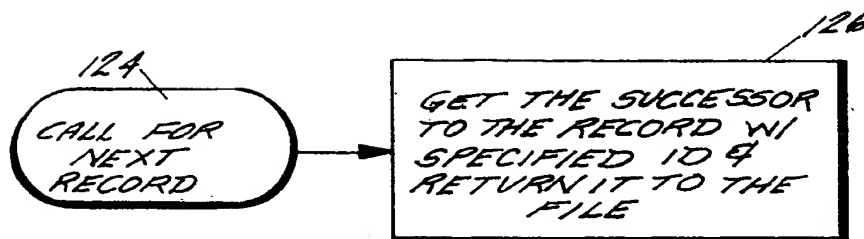


FIG. 10B

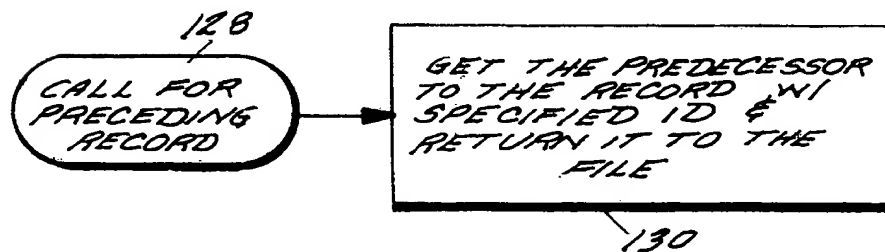


FIG. 10C

0278722

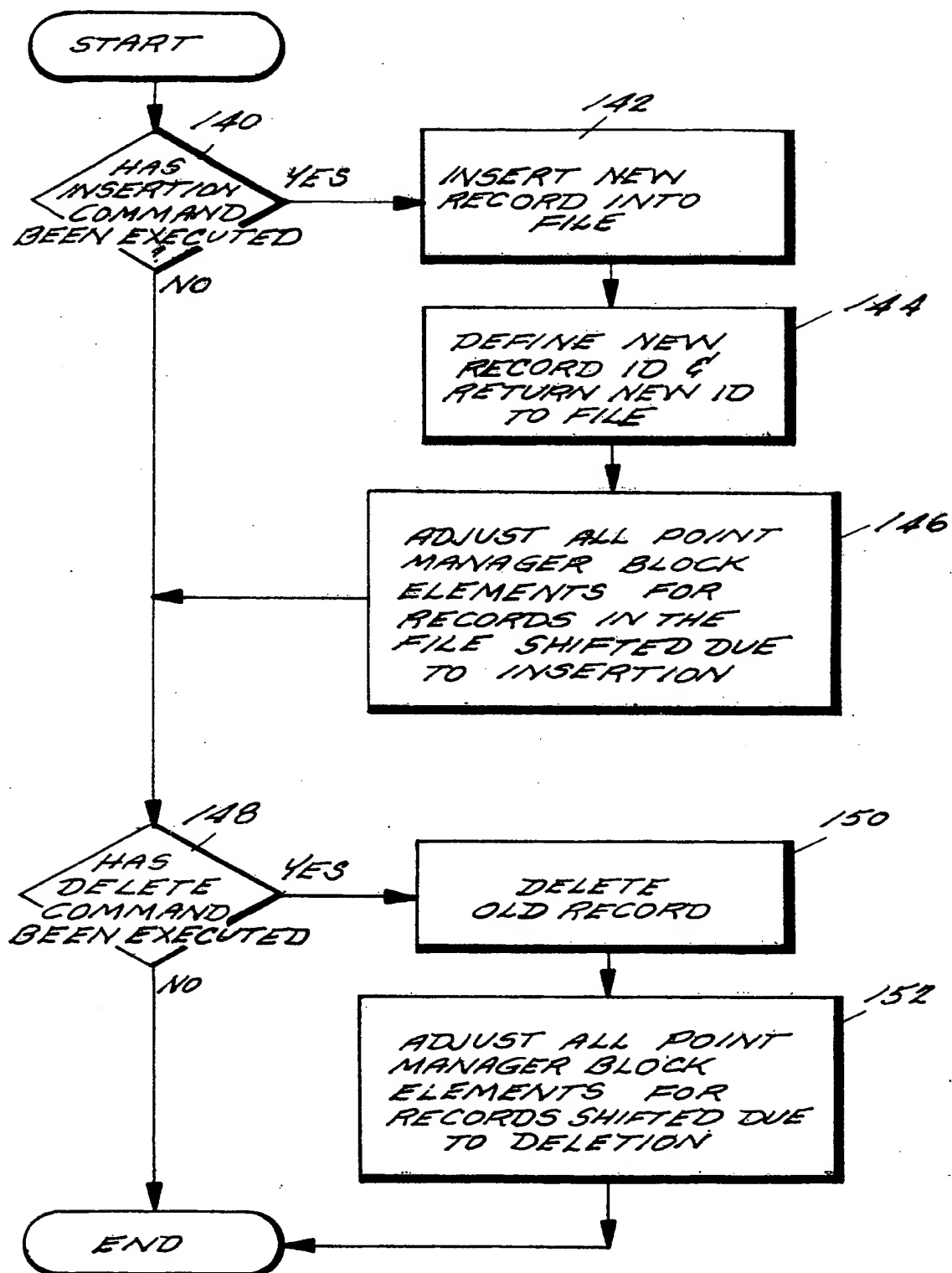


FIG. 10D

0278722

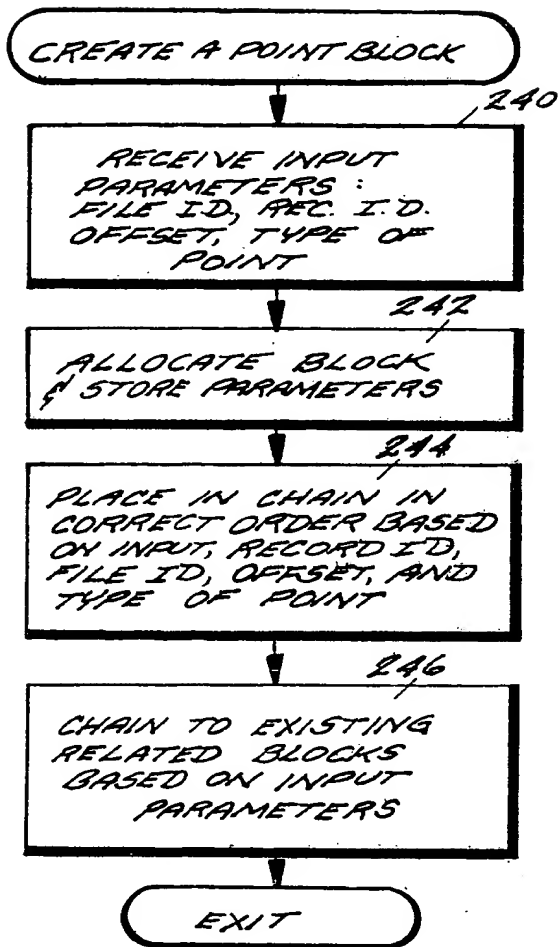


FIG. 12

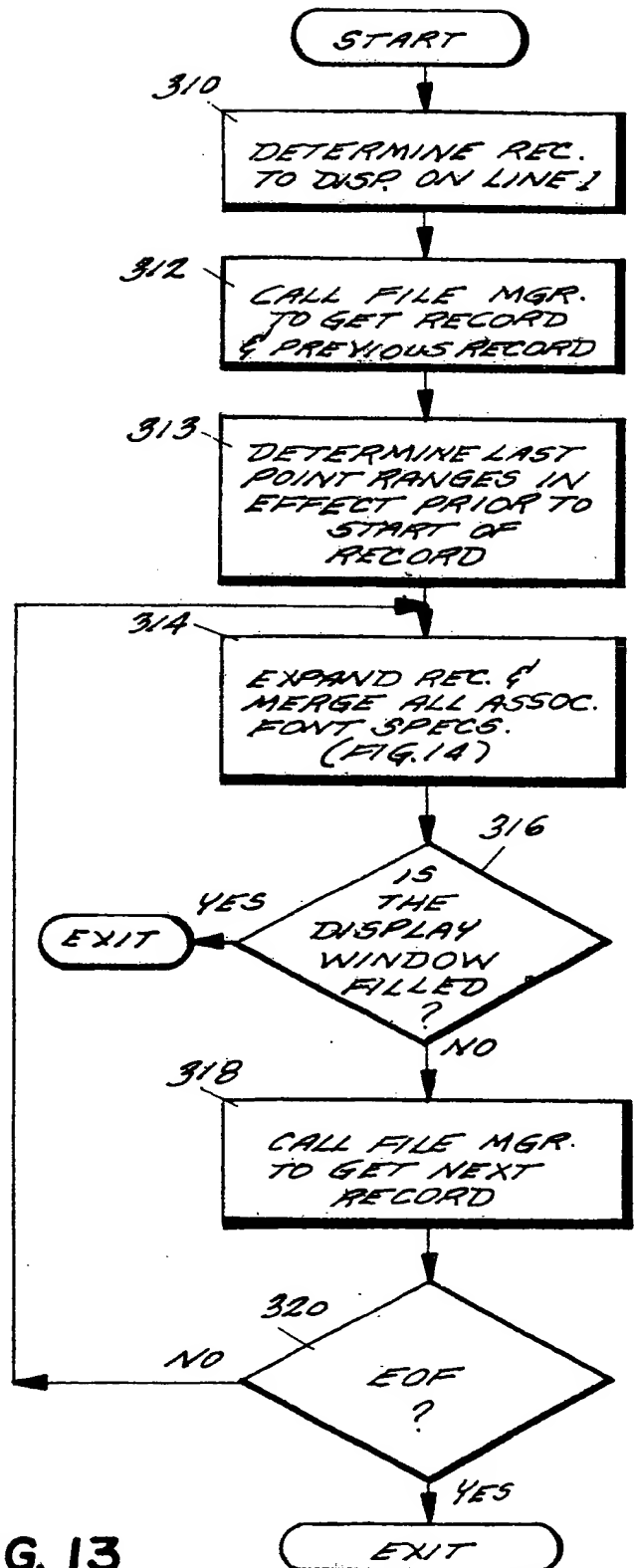


FIG. 13

0278722

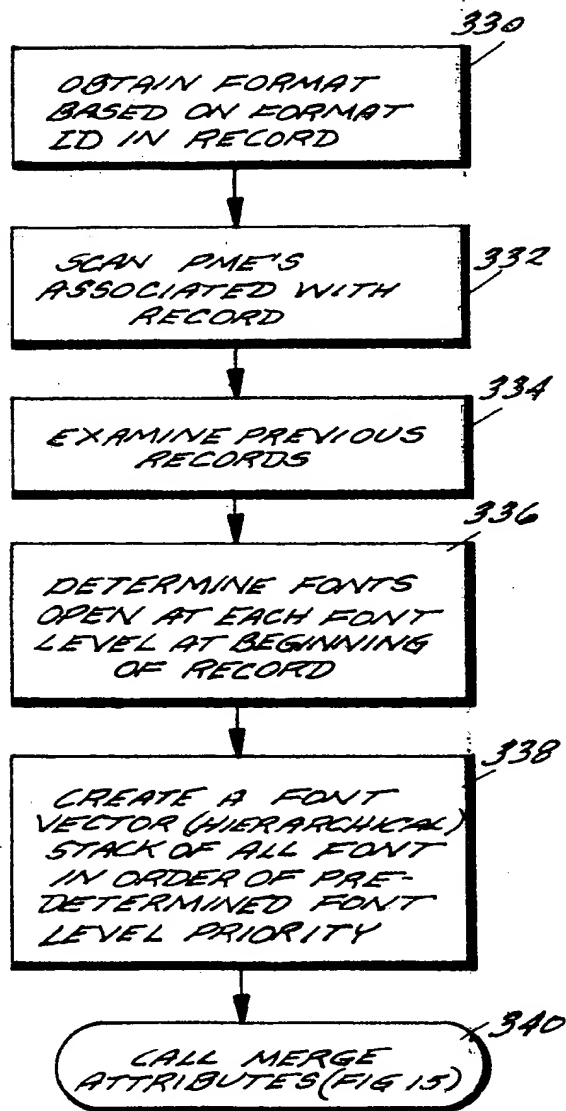


FIG. 14

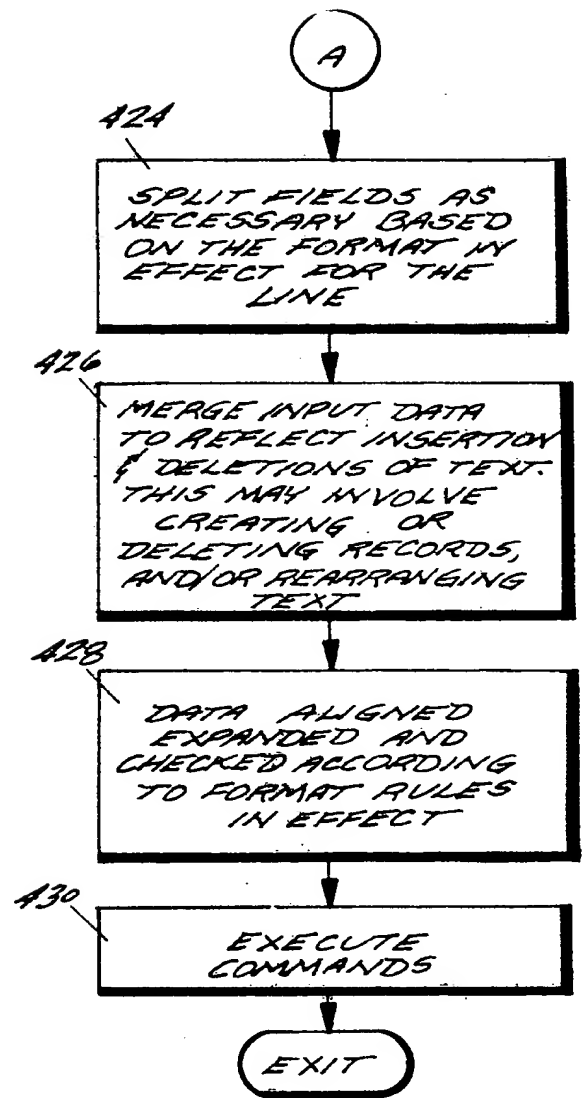


FIG. 17

0278722

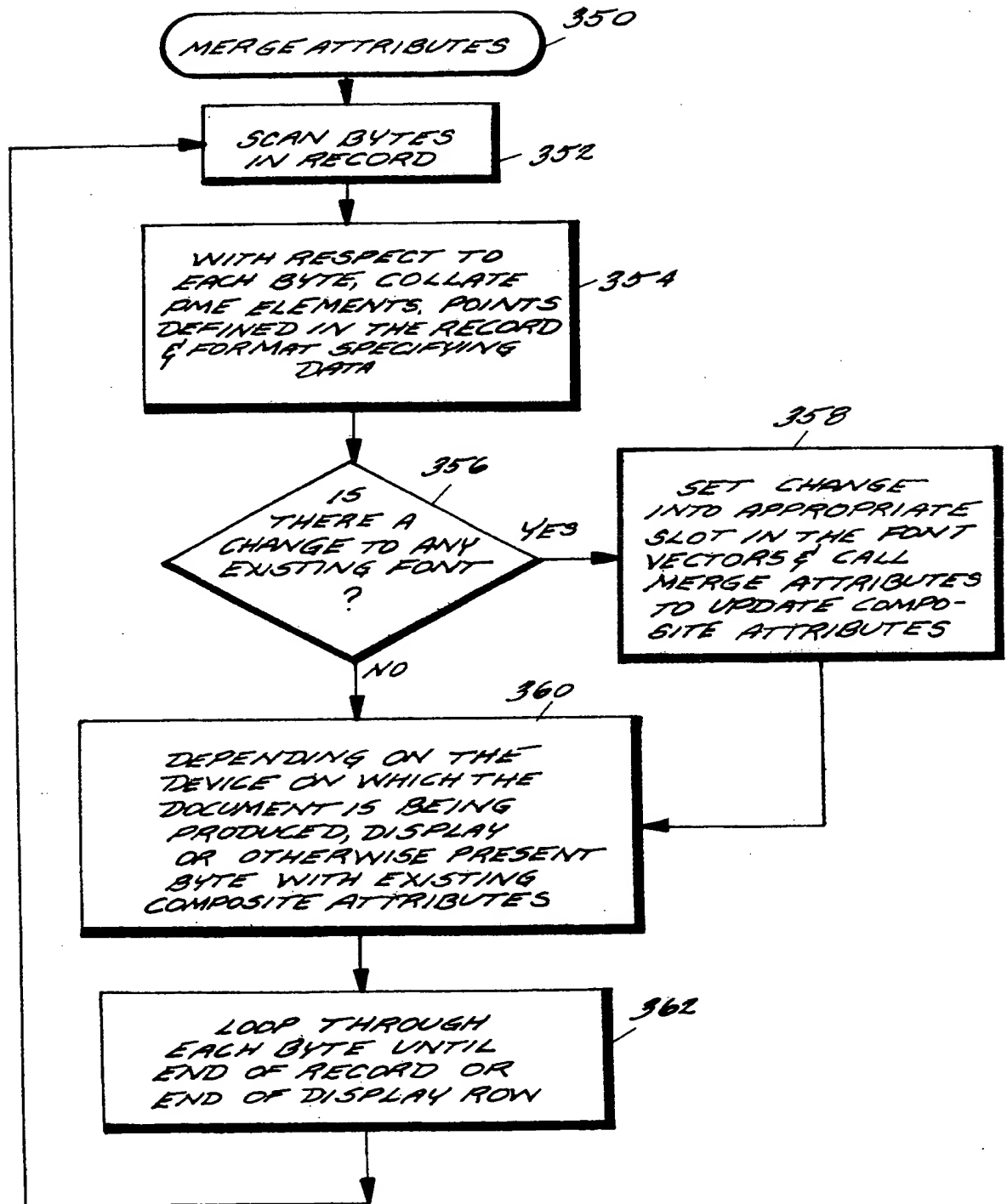


FIG. 15

FILE: MAPOLINE COPY A1 FISCHER-INNIS SYSTEMS CORP.

PAGE 016

```
ST      R0,MAPLRPSI      SAVE RESIDUE
B      TRIMOPUP          THEN TRIM LHS IF NECESSARY
DROOP  R14
SPACE 1
*****
*      ALL DATA DISPLAYED - FILL OUT LAST ROW
*
*****
SPACE 1
*** FILL OUT FINAL ROW WITH BLANKS OR NULLS
DATAWRN DS      OH      ALL DATA IS ACTUALLY WRITTEN
XC      MAPLRPSI,MAPLRPSI  RESET THE RESIDUE
L      R14,MAPLRPSI      OFFSET OF LAST ITEMAREA
A      R14,TEMAAPA      -> ASSOCIATED AREA
USING 1,TEMAAPA,R14
SPACE 1
NVI      MAPLRPSI,MAPLRPSI  USE BLANK IS ITEM PROTECTED
TM      MAPLRPSI,MAPLRPSI  IS ITEM PROTECTED?
BNZ     *+8             JUMP IF SO
NVI      MAPLRPSI,MAPLRPSI  UN-PROTECTED: EXTEND WITH NULLS
DROOP  R14
SPACE 1
*****
*      INSERT SPECIFIED CHARACTER TO END OF ROW
*
*****
CALL  XINDITE,VL=NO,PF=(E,MAPLRPSI),
      MAPLRPSI,MAPLRPSI  USE BLANK OR NULL
      MAPLRPSI,MAPLRPSI  MAXIMUM SIZE TO EXTEND
      MAPLRPSI,MAPLRPSI  WORK BUFFER OFFSET OF LAST ITEMAREA
      MAPLRPSI,MAPLRPSI  -> ASSOCIATED WINDOW
      MAPLRPSI,MAPLRPSI  INSUFFICIENT STORAGE?
      MAPLRPSI,MAPLRPSI  ERROR IF NO MORE STORAGE
      MAPLRPSI,MAPLRPSI
LTR     MAPLRPSI
BN      MAPLRPSI
SPACE 1
*** TRIM LEFT HAND SIDE WHICH MAY FAIL TO LEFT OF WINDOW
TRIMOPUP DS      OH
L      R0,MAPLRPSI      OFFSET OF FINAL ITEMAREA
L      R1,MAPLRPSI      -> ASSOCIATED WINDOW
CALL  TRIMLHS      TRIM THE LEFT-HAND SIDE
*      R15 = OFFSET OF FINAL ITEMAREA
L      R0,MAPLRPSI      LOAD RESIDUE OF UNDISPLAYED RECORD
SPACE 1
MAPLRPSI DS      OH
POPUP  RETR=(R15,R0)
SPACE 2
*****
DUMMYRDP DC      OH      DUMMY RPD USED TO SIGNIFY FORCED END *****
ORG      OA(0),XL(RPDELNGH),00
PC      RPDPOSN-RPDE+DUMMYRDP
DC      YL2(X'7FFF'),MAX POSSIBLE OFFSET
DC      AL1(PNTFINIS)
ORG      RPDELNGH+DUMMYRDP
SPACE 1
*****
DUMMYRDP DC      OH      DUMMY PME USED TO SIGNIFY FORCED END *****
ORG      OA(0),XL(PHDELNGH),00
PC      PHEOFFST-PHE+DUMMYRDP
DC      YL2(X'7FFF'),MAX POSSIBLE OFFSET
DC
```

0278722

30/46





THESE ARE PRESENTLY THREE TYPES OF PRECISE-POINTS:

1. - HARDWARE TAB  
1. - FORECAST SPECIFICATION
2. - NAME-RE-LABEL  
2. - RPD: IGNORED (THE DATA IS ALSO STORED IN THE PHE)
3. - INACTIVE (HISTORICAL) INSERT-POINTS  
3. - PHE: THESE ARE PREVIOUS INSERT POINTS TO WHICH THE USER MAY WISH TO AUTOMATICALLY RETURN.

THE ONLY CASE IN WHICH SPECIAL WORK IS REQUIRED IS FOR A  
HARDWARE TAB - WE SET A SPECIAL FLAG TO FORCE AN INSERTED  
ATTRIBUTE BYTE.

```

PUNCT005 DS      OH,=Y(*WITHPMT) FROM FORMAT SPECIFICATION?
CH          BNE      PROCESS3(R4) NO - CANNOT BE HARDWARE-TAB
SPACE 1    L        R14,MAPLWIND -> ASSOCIATED WINDOW
L          USING    WINDOWAREA,R14
TH         WINDFELS,WINDPROT+*INDPROT+*INDQUAL VIEW PROTECTED OR QUALIFIED?
BNZ        PROCESS3(R4) YES - AVOID INSERTING HARDWARE TAB
DROP      R14
SPACE 1    TO      INSERT HARDWARE TAB ON NEXT CALL TO 'MAPFIELD'
O1         MAPLTAB,MAPLHTAB SHOW HARDWARE TAB PRESENT AT POINT.
B          PROCESS3(R4) (NEW FIELD ALREADY STARTED)
SPACE 1
**** SET

```

FORMAT-FIELD END

IF PRESENT IN RPDE THEN THIS WILL BE ENCOUNTERED TWICE  
FOR THE SAME SPECIFICATION -  
- ONCE IN THE RPDE  
- AND AGAIN IN THE SPECIALLY MODIFIED PMT VECTOR.

HOWEVER, DUPLICATE ENCOUNTERS ARE NOT BE A PROBLEM, SINCE THEY OCCUR WITH THE SAME INFO AND THE SAME BYTE.

THIS DOES NOT ACTUALLY CAUSE A NEW FIELD TO BE STARTED, ALTHOUGH IT WILL AUTOMATICALLY START NULL INSERTION (UNLESS THE FOLLOWING CHARACTER IS A REQUEST FOR BLANK FILLING). THIS FORMAT ATTRIBUTE IS MAINLY APPLICABLE WHEN THE LINE IS BEING FORMATTED.

```
*****  
**  
** FUNCTION06 DS CH PROCESS3(R4) ADVANCE TO NEXT SPECIFICATION,  
** TITLE : MAPOLINE - *****  
** ADVANCE JUST-PROCESSED SPECIFICATION POINTER *****  
**
```

```
*****  
+ + +  
PROCESS3 VECTOR1  
SPACE 1  
ADVANCE EATPOINT ELEMENT  
{WITHPT, NEXTPT}  
{WITHRPD, NEXTRPD}  
{WITHPHE, NEXTPHE}  
SPACE 2  
***** ADVANCE TO NEXT FORMAT POINTER *****  
NEXTPT DS OH FPNTRNGH(, R7) ADVANCE TO NEXT PNT POINTER ELEMENT  
LA B FROMTOP  
SPACE 1  
***** ADVANCE TO NEXT RECORDED-POINT ELEMENT *****  
NEXTRPD DS OH RPDELNGH(, R6) ASSUMP THERE IS ANOTHER RPDE  
LA LA R6 MAPLRPDZ DOES THIS EXCEED RPDE VECTOR?  
BL PROMTOP JUMP IF SO  
LA R6 DUMHYRDP ELSE USE DUMMY RPD  
B FROMTOP  
SPACE 1  
***** ADVANCE TO NEXT POINT-MANAGER ELEMENT *****  
NEXTPHE DS OH RETAIN ADDRESS OF CURRENT PHE  
LH L14, R5 OFFSET OF NEXT PAR  
L R5, PHENEXT HAVE WE REACHED THE END?  
LTR P5, R5 JUMP IF SO  
BNP USEDUMMY -> ACTUAL ASSOCIATED PHE  
A R5, TERAPHA  
SPACE 1  
***** INSURE NEXT PHE IS FOR SAME RECORD *****  
CLC PERECID, PHRECID-PHE(R14) NEXT PHE FOR SAME RECORD?  
BNE USEDUMMY SAME FILE?  
CLC PREFILE, CHEPILE-PME(R14) YES - THEREFORE POINT FOR SAME RECORD  
BE FROMTOP  
SPACE 1  
DS OH USEDUMY DS OH OTHER PMES EXHAUSTED  
LA LA B PROCESS WITH DUMMY PHE  
B FROMTOP  
SPACE 1  
***** DISPLAY GROUP FULL - SOME RIGHT-HAND DATA NOT DISPLAYED *****  
NODATAI DS OH COME HERE WHEN NO REAL DATA DISPLAYED  
SH R1, R1 ROW IS FULL  
SPACE 1  
***** R1 = NUMBER OF DATA BYTES EFFECTIVELY DISPLAYED IN CURRENT SET *****  
LINEDONE A R1 MAPLCURR GIVES OFFSET BEYOND DISPLAYED DATA  
DS A R14 MAPLPBCA -> RECORD  
L USING RECORD, R14 SIZE OF THE RECORD'S TEXT  
LH R0, RECCEXTZ THIS GIVES NON-DISPLAYED RHS RESIDUE  
SR R0, R1
```

[illegible]

0278722

27/46

PAGE 013

75

FILE: NAPOLINE COPY A1 FISCHER-INNIS SYSTEMS CORP.

```

R14=X(PFLDLENCH)  OFFSET INTO FIELD VECTOR
R14=PNTFIELD-THIS PFC-PFLDLENCH(R14) OFFSET OF PNTFIELD
A  GET ABSOLUTE POINTER
R14,HAPLPNTA  SAVE CURRENT PNT FIELD
ST  USING PNTFIELD,R14
SPACE 1
*   ALSO SAVE PNT WHICH STARTS WITH THIS FIELD
LH  R0,PFLDLENCH  GET PNT IDENTIFIER FOR THIS FIELD
L  R15,HAPLPONT  -> PNT VECTOR
STH R0,2*PONTPTNT(R15) SET NEW PONTPTNT PNT LEVEL
DROP R14
HVC NULLPILL=Y(MAX)  RESET NULL FILLING
B  PROCESS3(R4)
TITLE,NAPOLINE-PROCESS 'TARGET' POINT'
*****
*   PROCESS 'TARGET' POINT
*****
*   JSE 'PNT' INFO TO CONSTRUCT PROPER CONTROL-POINT FOR THE TARGET
*   POINT. IT WILL BE RESTORED AFTER THIS SINGLE POINT.
*****
TARGETXX DS
L  R14,HAPLPONTV  -> PNT VECTOR
LH R2,2*PONTCNTL(R14)  RETAIN CURRENT CONTROL POINT
LH R0,PHPONTNTI  GET PNT FOR TARGET VALUE
STH R0,2*PONTCNTL(R14)  SET REVISED PNT
SPACE 1
CALL HAPTPFLD,VL=NO,MF=(E,HAPLNORK),
      (HAPLPONTV),
      HAPLPARC,
      HAPLPARC+1,
      =A(1),
      =A(0),
      =AL(ITEMPROT+ITEMMARE),ROWS IF PERMITTED
      <HAPLXROW>,
      <HAPLWIND>,
      HAPLAST,
      =AL(0)
      NO HARDWARE TAB
*****
SPACE 1
**** RESTORE PREVIOUS CONTROL VALUE, BHPQP CONTINUING
L  R14,HAPLPONTV  -> PNT VECTOR
LH R2,2*PONTCNTL(R14)  RESTORE PREVIOUS CONTROL POINT
STH R15,R15  INSNRP ALL IS WELL
BM HAPLO800  EXIT IF WE HAVE AN ERROR
LTH R0,R0  IS REQUESTED DISPLAY ENTIRELY FULL?
LTH R0,R0  NO - CONTINUE PROCESSING
B  PROCESS3(R4)  ELSE EXIT IF DATA DISPLAYED BEFORE
      NODATAAXY  THE RECORD IS DONE.
*****
TITLE,NAPOLINE - PRECISE-POINT DEFINITION
*****
PRECISE-POINT DEFINITION
*****

```

```

0=A(1),
MAPLCURR,
=AL1(ITEMKISRT),
=AL1(ITEMKISRT),
<MAPLXROW>,
<MAPLWIND>,
MAPLAST,
=AL1(01)
R15,P15
MAPL0800
PO,RO
NOBATAXY
LTR
BN
LTR
RM
NO AUXILIARY CODE-POINTS
TRY TO INSERT SINGLE BYTE FIELD
CURRENT RECORD OFFSET
SHOW THIS IS 'INTERPRET' FIELD
MAXIMUM ROWS ALLOWED FOR THIS LINE
--> WINDOW
--> WORD WITH OFFSET OF LAST ITEMAREZA
NO HARDWARE TAB
PERMANENT ERROR?
ERROR IF SO
NUMBER OF AVAILABLE COLUMNS IN ROW
PIXI IF FINAL ROW IS FINISHED

```

```

**** IF ZERO BYTES ACTUALLY PROCESSED, THEN WE KNOW
* THE FIELD IS PROTECTED AND INSERTION IS IMPOSSIBLE.
  LCR R1,R1 WAS ANYTHING INSERTED?
  RZ PROCESS3(RB4) INSERTION DISALLOWED,
  SPACE!

```

\* \* \*

SPACE !

\* \* \*

FILL OUT LATEST ITEM WITH BLANKS/PULLS

\* \* \*

```
*****
*      SPACE 1
*      P2 = NUMBER OF AVAILABLE COLUMNS IN ROW
*      E1 = NEGATIVE OF NUMBER OF BYTES JUST INSERTED
*      R15 = NUMBER OF ROWS RECOMPASSED BY THIS ENTIRE LINE
```

\* \* \* \* \*

\*\* DETERMINING IF WE SHOULD WF FILL OUT THE ROW TO THE END

FILLOFF DS OH R1,MAPLI\*MT DEMICT NUMBER BLANKS JUST INSERTED  
 A ST R1,MAPLIANT THIS GIVES COUNT YPT-TO-INSERT  
 CB R1,RO YET-TO-INSERT EXCEED BALANCE IN ROR?  
 RVL USPRECS YPS - SO EXPRESS FULL REQUEST

```

* ONLY PARTIAL LINE REQUESTED - SHOULD WE FILL IT ALL THE WAY OUT?
  SPACE 1
  L H4,MAPL302
  CL R15,0(,R14)
  RM USRP05T
  ARE WE CURRENTLY ON LAST R02?
  YES - DO NOT BE TOO GENEROUS W/SPACE
  ARE WE CURRENTLY ON LAST R02?
  --> WORD WITH ROW-LIMIT
  - SHOULD WE FILL IT ALL THE WAY OUT?

```

```

SPACE 1
TM MAPLPLGS,MAPLLEOR SHOULD WE INSERT TO END-OF-ROW?
      (SET WHEN FMTFIELD EXAMINED)
      YES - SO USE FULL ROW
      ELSE REQUEST FULL ROW
B2 USERQST
IR R1,RQ

```

```

*****
SPACE 1
*****
*****
INSERT (PSEUDO-) NULLS AS NEEDED
*****
*****

```

```

SPACE 1
*** REQUEST FULL AMOUNT OF INSERTION (EVEN IF IT EXCEEDS ROW)
* R1 = NUMBER OF BYTES TO REQUEST TO INSERT INTO THIS ROW

```

JSEREQST DS OR

ВГР: НА ПОЛІТНЕ СОРУ

```

+ + + + +
R2,R1      NUMBER OF BYTES TO INSERT
XINDITEM,VL=NO,MPI=(E,HAPLSOBRK),
(BAPLSUDOC,
(R2)       MAXIMUM SIZE TO EXTEND
HAPLAST    WORK BUFFER OFFSET OF LAST ITEMAREA
<HAPLBIND>) -> ASSOCIATED WINDOW
R15,R15    SUCCESSFUL?
HAPLO800   NO - INSUFFICIENT STORAGE
PROCESS3(R4) EXIT WHEN INSERTION IS COMPLETE
SPACE 1
R15,HAPLAST SAVE AMOUNT YET TO BE INSERTED
ST STARTINS YES,INSERT UNTIL EXHAUSTED
B TITLE,HAPOLINE - FONT-RANGE START/END.
*****
+ + + + +
FONT-RANGE START
*****
+ + + + +
RPDE - DEFINES BASE MODIFIER OR HIGHLIGHT FONT
PME - DEFINES CONTROL FONT.
*****
+ + + + +
FUNCTION04 DS OH O(4(F4)
          B VECTABLE MAX=WITHMAX,
           { WITHPNE,F004XPHE},
           { WITHPRD,F004XRPD})
*****
+ + + + +
SPACE OH
F004XRPD DS R1,RPDMULTI
          LH R15,HAPIFILE
          L USING FILEDEFN,E15
          L DROP R15,FILEPONV
          LH R15,OJ(E1,E15)
          B F002XRPD
*****
+ + + + +
SPACE 1
*** PROCESS POINT-MANAGER ELEMENT
F004XPHE DS OH
          LH RO,PHEFONTI
          B F002XPHE
*****
+ + + + +
SPACE 2
*****
PCNT-RANGE END
*****
+ + + + +
RPDE - DEFINES BASF, SPECIFICATION OR HIGHLIGHT FONT
PME - DEFINES CONTROL FONT.
*****
+ + + + +
FUNCTION02 DS OH
          SR RO,FO
          B <4(IRU)
          B VECTABLE MAX=WITHMAX,
*****
WE ARE CLEARING FONT-INDEX VALUE
PICK UP SOURCE OF THE POINT

```

0278722

22/66

PAGE 038

70

FILE: MAPOLINE COPY A1 FISCHER-INNIS SYSTEMS CORP.

```

*****
*
* 1. GIVEN "N" = INSERT COUNT (ORIGINALLY SPECIFIED ON
*     THE INSERT COMMAND, AND NOW CARRIED IN THE PHE),
* 2. INSERT SIZE = MAX (20, N * CURRENT-FIELD-SIZE),
* 3. IF THIS DOES NOT OVERFLOW THE CURRENT ROW, THEN THIS
*     IS EXACTLY THE SIZE.
* 4. IF THIS DOES OVERFLOW THE ROW, THEN WE MAY EXTEND THE
*     INSERTION RANGE FURTHER TO THE END OF THE CURRENT IP:
*     A. THE FIELD IS WRAPPABLE (LAST FIELD IN THE LINE), AND
*     B. THE INSERTION DOES NOT EXTEND TO LAST ALLOWED ROW.
*
*****
*
* *****
*
* SPACE 1
*
* *** IGNORE INSERT IF:
*   . THIS INSERT POINT IS INACTIVE
*   . THE CURRENT ITEM IS PROTECTED
*   . THIS IS NOT THE FINAL FIELD IN THE LINE
*   . FIELD IS MARKED NOT-INSERTABLE
*
*****
*
* INSERTYX DS OH
* MAPLEGS,X'FF'-MAPLEOF RESET THE INSERTION FLAGS
* LA R3,4*PNT*EXACT THIS RECORD'S PRECISE-POINT
* TM PMETVEE,P*REACTIV IS THIS PME ACTIVE?
* BZ TRUNCPRV NO - SO SIMPLY USE ATTR TO HOLD POS.
*
* *****
*
* SPACE 1
*
* *** DETERMINE SIZE OF THE LOGICAL FORMAT FIELD
* L R15,MAPLEMTA -> ACTIVE FORMAT HEADER
* USING R15,PREC,R15
* L R14,R14,MAPLEMTIC
* LTR R14,R14
*
* *****
*
* DE TRUNCPRV
* USING FMTFIELD,R14
* CLC FFLDINDX,FMTFLDCT+L,FMTFLDCT-L,FFFLDINDX LAST FIELD?
* RNE **8 JUMP IF NOT
* B **8
*
* MNOTE 1 'WE MAY NOT HAVE THE PRECEDING COMPARE CORRECT'
* OI MAPLEGS,MAPLEFFIF SHOW THIS IS FINAL FIELD IN FORMAT
*
* *****
*
* SPACE 1
*
* *** DETERMINE SIZE OF THE FIELD
* TM FFLDATTR,FFFLDWRAP
* BZ **8
* OI MAPLEGS,MAPLEFFIF SHOW THIS IS FINAL FIELD IN FORMAT
*
* *****
*
* MNOTE 2 'FOLLOWING CHECK ONLY FOR VERSION 1 - UNTIL WE SUPPORT+
* FORMAT "STRETCHING" DOES FIELD SUPPORT WORD WRAP?
* TM FFLDATTR,FFFLDWRAP
* BZ TRUNCPRV NO - THEREFORE AVOID INSERTION
*
* *****
*
* SPACE 1
*
* LH R0,FFFLDEND OFFSET BEYOND END OF FIELD
* SH R0,FFFLDSTRT GIVES LENGTH OF FIELD
* ERROR NP WHAT HAPPENED HERE??
* DEOP R14
* SPACE 1
*
* *****
*
* *** SET INSERTION COUNT

```

0278722

23

66

71

PAGE 039

FILE: HAPOLINE COPY A1 FISCHER-INNIS SYSTEMS CORP.

```

R15,PHINSRT      COUNT OF 'LINES' TO INSERT
R15,R15          INSURE POSITIVE
BP 008          NUMBER MUST BE POSITIVE
LA R15,1
SPACE 1
*** COMPUTE SIZE OF INSERTION
R14,R0          THIS GIVES BYTE COUNT
R15,R15,Y(20)   ALWAYS TAKE AT LEAST 20 BYTES
BNL 008
LA R15,20       ELSE USE MINIMUM VALUE
ST R15,HAPLIANT SAVE INSERT BYTE COUNT FOR LATER
SPACE 1
*** DETERMINE IF WE HAVE UNCLOSSED LEFT-CONTIGUOUS DATA
CLC             DETERMINE IF CURRENT = PREVIOUS
R15,HAPLCORP,HAPLPREV YES - NO DATA, INSERT IS NEW FIELD
BE STARTINS     SHOULD NEVER EVER HAPPEN
PROR 1
SPACE 1
*** THIS IS INSERTION WITH DATA TO THE LEFT
LA R15,40PNTAPPND SET FOR RETURN TO 'APPEND' ROUTINE
B TRUNCPRV      AFTER HE TRUNCATE THIS DATA.
SPACE 1         TRUNCATE PREVIOUS DATA
*** RETURN AFTER INSERT TO AUGMENT LEFT-CONTIG DATA
SPACE 1
*** REGISTERS AS ON RETURN FROM 'HAPFIELD':
R0 = NUMBER OF VISIBLE BYTES REMAINING IN ROW
R1 = NUMBER OF BYTES ABSORBED
R15 = NUMBER OF POES ENCOMPASSED BY ENTIRE LINE
*** TRY TO AUGMENT PREVIOUS FIELD BY ADDING RIGHT-HAND NULLS.
AFTRINST DS     OFFSET OF CURRENT FINAL ITEMAREA
L R14,HAPLIANT  -> MOST RECENTLY ADDED ENTRY
A R14,TERMAPA
USING I,TERAREA,R14
SPACE 1
R1,R1          PREPEND NO INSERTION AREA YET ADDED
R1,TYPE,ITEMPROT IS THIS PROTECTED?
TH R1,ITEMPROT NO - THEREFORE INSERT AT END OF DATA
BZ FILLOUT
DROP R14
SPACE 1
*** INSERTION STARTS/RESUMES AT NEXT AVAILABLE FIELD
SPACE 1
STARTINS DS     HAPFIELD,V1=NO,WF=(E,HAPLWORK),
CALL            -> CALLER'S POINT VECTOR
                (<HAPLFONV>,'
                HAPLSUDO, -> PSEUDO-BLANK

```



0278722

20/46

FILE: MADOLINE COPY AT FISCHER-INNIS SYSTEMS CORP.

B4E  
 WHICH  
 P4 = Y(4\*WITHPMT)  
 PROCESS3(B4)  
 IS THIS ACTUALLY 'FORMAT' SPEC?  
 NO - TAKEN CARE OF BY NEW PHT ITEM

```

**** DETEINE
SR      INDICATOR OF CURRENT FORMAT FIELD
R14,R15    R14, R15    FORMAT FIELD INDEX
IC      R14, R15    OFFSET INTO FIELD VECTOR
MH      R14, R15    OFFSET OF FMTFIELD
LA      R14, R15    GET ABSOLUTE POINTER
A       R14, R15    GET ABSOLUTE POINTER
USING   FMTFIELD.R14

```

\*\*\*\*\* IS THIS FORMAT FIELD ALREADY UNDERWAY?  
C R14,MAPLFTHC FIELD ALREADY STARTED?  
DE PROCESS3(R4) YES - SO IGNORE THIS POINT

```
*
RECEIVE      ! PARAGRAPH START ONLY IF START-OF-PARAGRAPH
L            -> RECORD
USING        R15,RAPLRCA
              RECD,R15
TH           RECTYPE,RECPARAS
BZ          PROCESS,(R4)
B           TRNCPRVN
              IS THIS PARAGRAPH START?
              NO - SO IGNORE THIS POINT
              TREAT THIS LIKE FORMAT-START
```

```
*****
TITLE 'NAPOLINE - TRUNCATE PREVIOUS FIELD' *****
*****
* * *
TRONCATE PREVIOUS FIELD
*****
```

```

* B2C:EV DS
IP LENGTH OH
IS NOT POSITIVE, THEN NOTHING TO TRUNCATE
CURRENT OFFSET LESS
PREVIOUS OFFSET = SIZE TO DISPLAY
NILL - NOTHING TO TRUNCATE
ERROR IF LENGTH IS NEGATIVE
SIZE TO BE DISPLAYED
* R0,MAPLWS12
R0,MAPLCURR
R0,MAPLSREV
PROCESS2 (R3)
ERROR N
R0,MAPLWS12
ST

```

```

***** INVOICE MAPFIELD TO TRUNCATE THIS ITEM
L USING RECORD, R14 -> START OF THE RECORD
SR R2, R2 ASSUME NO AUXILIARY TEXT
LA R15, RECTEXT -> START OF TEXT
SPACE 1
TN RECFLAGS, RPCGRAFX AUXILIARY GRAPHICS STRING?
BZ *+10 JUMP IF NOT
LN R2, RECTEXTZ SIZE OF EACH TEXT SEGMENT
AP R2, P15 -> START OF SECONDARY AREA
DROP R14
ST P15, MAPITEMP THIS IS TEMPORARY POINTER

```

```

**** CJT THE CURRENT FIELD & MOVE TO ITEXTAREA BUFFER
CALL MAPFIELD_VL=NO,MF=(P,MAPFORK),
      (<NAPLPONT>,
       --> PRIMARY DATA
        <NAPLTMP>,

```









0278722

17/46

65

FILE: HAPOLINE COPY A1 FISCHER-INNIS SYSTEMS CORP.

```

*****
SPACE 1
*** GET ADDRESS OF UNDERLYING FORMAT
LA R0,RPDECD
R1,HAPLPFILE
CALL GETFORMAT
LTR R7,R15
BNP HAPLO800
USING PHISPEC,R7
SPACE 1
ST R7,HAPLFMTA
ST R0,HAPLPMT0
SPACE 1
*** SET POINTER TO STANDARD FORMAT POINT VECTOR
AU R7,PMTPOFF
USING PMTPOINT,R7
SPACE 1
*****
b
*
*
*****
ACCESS RECORD POINT DEFINITION VECTOR
*****
*
*
*****
SPACE 1
COMPUTE END OF RPDE VECTOR
LA R14,RECSIZE
LA R14,RECORD(R14)
ST R14,HAPLRPDZ
SPACE 1
LOCATE START OF THE RECORDED-POINT VECTOR VECTOR
LA R6,RECTEXT2
T4 RECTEXT2
B2 *+6
AR R6,R6
LA R6,RECTEXT(R6)
SPACE 1
CR R6,R14
BL *+6
LA R6,DONHYPD
USING RPDE,R6
SPACE 1
*****
*
*
*****
ACCESS POINTER-MARKERS FOR THIS RECORD
*****
*
*
*****
LOCATE START OF POINTER-MARKERS FOR THIS RECORD
L R5,HAPLPFILE
USING FILEDEFN,R5
CALL PHAPIND,61=NO,MP=(E,HAPLMORK),IDENTIFIER
(FILESELF,
<HAPLRECI>,
=A(0))
START WITH OFFSET = 0
*****
*
*

```

```

-> RECORD FOR WHICH WE WANT THE PNT
-> FILEDEFN
GET ADDRESS OF (PINNED) FORMAT BLOCK
ADDRESS OF TRUE PHISPEC
EXIT IF NOT

```

```

SAVE ADDRESS OF EFFECTIVE 'PHISPEC'
SAVE ADDR OF POSS ORIGINAL PNT IMAGE
-> START OF POINT VECTOR

```

```

SIZE OF ENTIRE RECORD
-> BYTE FOLLOWING RECORD
(WHICH FOLLOWS RPD VECTOR).
SAVE END OF RPDE LIST

```

```

ALLOP FOR EXTENSION
-> START OF RECORDED POINT DEPNS
DOES VECTOR EXIST?
JUMP IF SO
ELSE USE DUMMY 'TERMINATING' RPDE

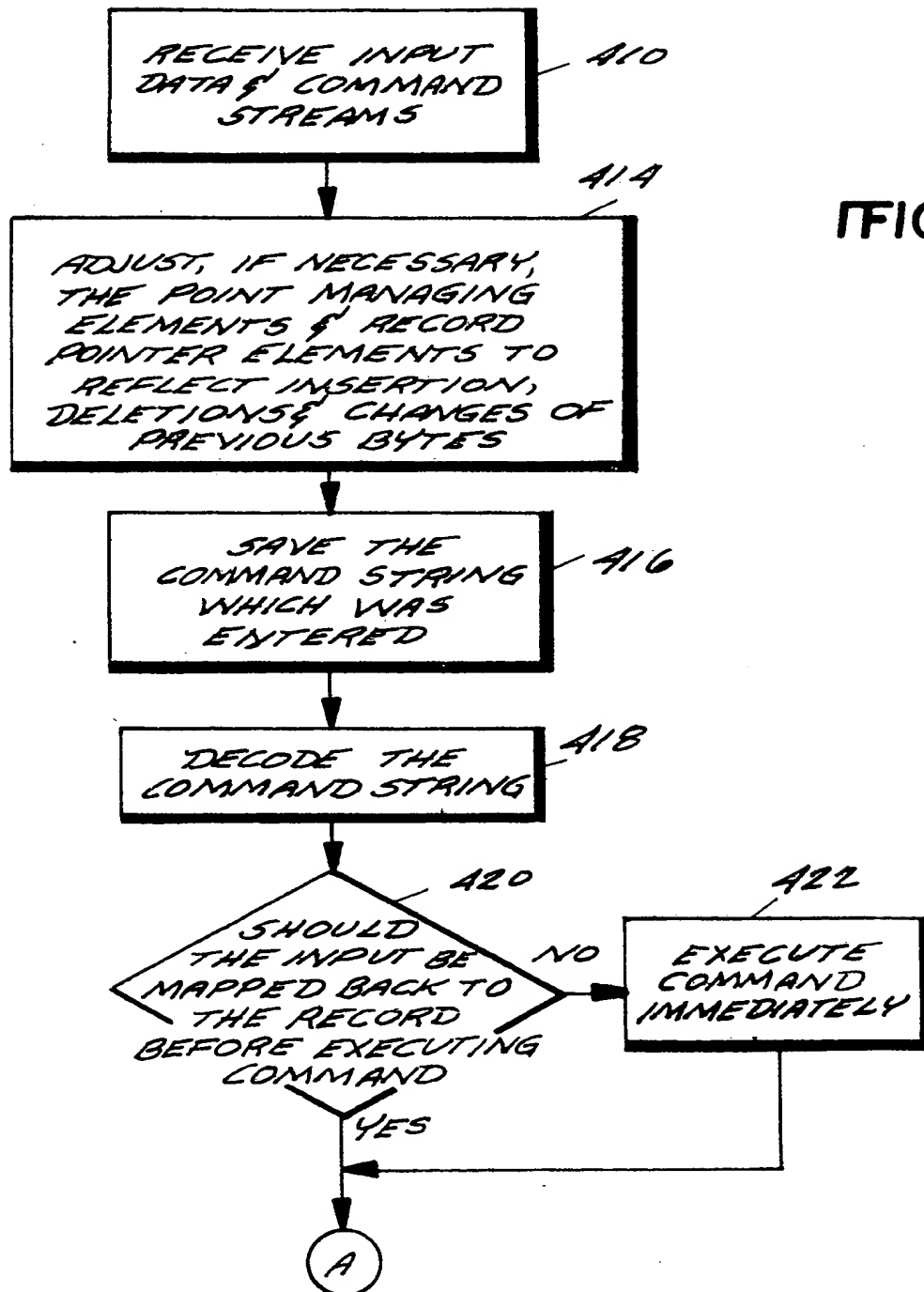
```

```

-> ASSOCIATED FILE DEFINITION
(RSN, FILE, IDENTIFIER)
-> RECORD IDENTIFIER
START WITH OFFSET = 0

```

0278722



## 63

$$15 \overline{) 46}$$

```
***** TITLE 'MAPOLINE' - MAP OUTPUT LINE # OF DISPLAY PURPOSES *****  
***** M A P O L I N E *****  
***** MAP OUTPUT LINE FOR DISPLAY PURPOSES *****  
  
DESCRIPTION -  
THIS ROUTINE MAPS AN OUTPUT RECORD ONTO INTO AN  
ITEMAREA VECTOR IN PREPARATION FOR ULTIMATE DISPLAY.  
  
A NEW ITEMAREA VECTOR IS CONSTRUCTED IN A TEMPORARY  
HOLDING AREA. THE FINAL RESULT IS THEN COMPARED  
WITH THE EXISTING DATA, IF ANY, AND THE FIELDS WHICH  
ARE CHANGED ARE MARKED FOR EVENTUAL 3270 DISPLAY.  
  
***** INPUT - R1 *****  
-> PARAMETER LIST  
+00. -> RECORD IMAGE TO BE EXPANDED FOR DISPLAY  
+?? -> RECORD IDENTIFIER (B/E)  
+08 -> INPUT FONT VECTOR (WILL BE MODIFIED ON EXIT)  
+12 -> WORD WITH ROW COUNT LIMITATION  
+16 -> WINDAREA (PINNED)  
+20 -> FILEDEFN (PINNED)  
-> TEST  
  
***** K12 *****  
  
***** OUTPUT - R15 *****  
>=0 => P15 = OFFSET (IN TEMAMAPA) OF LAST ITEMAREA;  
=-1 => RO = NUMBER OF UNDISPLAYED RHS RECORDED BYTES.  
=-2 => UNSUCCESSFUL STORAGE  
  
COPYRIGHT 1986, ADDISON FISCHER - ALL RIGHTS RESERVED.  
  
SPACE 1 ***** DEFINE PARAMETERS LIST *****  
PUSHSECT ,  
MAPLPARM DSECT ,  
MAPLAREA DS  
MAPLRPCI DS A  
MAPLPDND DS A  
MAPLXCON DS A  
MAPLTAND DS A  
MAPLFLE DS A  
SPACE 1  
MAPLAREA DSECT f  
MAPLCURR DS  
MAPLPREV DS  
MAPLOSDV DS  
MAPLAST DS  
MAPTEMP DS  
MAPPLSIZ DS  
MAPPLANET SPACE 1
```



0278722

32  
46

80

PAGE 002

FILE: GATHFONT COPY A1 FISCHER-INNIS SYSTEMS CORP.

GATHFONT PUSHDOWN BASE=R11,SAVE=YES,TYPE=STACK, LINKAGE  
WORK=(GATHLNGH,GATHAREA,R10)

```

SPACE 1
USING GATHPARM,R1
      R14,GATHRECA
      GATHRPCI,0(R14)
      R7,GATHVECT
      R8,GATHVIEW
      VIEWDEFN,R8
      DROP R1
      USING R6,TEPHAPXA
      USING PNE,R6
      SPACE 1
      $USING FILEDEFN,R5),FROM=VIEWFILE
      R3,VIEWFILE
      RSH FILEDEPN TOKEN
      SPACE 1
      XC 2*FONTATTN(2,R7),2*FONTATTN(R7)
      XC 2*FONTCTL(2,R7),2*FONTCTL(R7)
      SPACE 1

```

```

      -> RECORDED IDENTIFIER
      SAVE RECORDED IDENTIFIER
      -> ASSOCIATED FONT VECTOR
      -> (PINNED) VIEWDEFN
      -> (PINNED) POINT MARKER INDEX

```

```

      2*FONTATTN(2,R7),2*FONTATTN(R7)
      2*FONTCTL(2,R7),2*FONTCTL(R7)
      ASSUME ATTENTION AND
      CONTROL LEVELS NULL

```

```

*****
LOCATR ATTN POINT MOST CLOSELY PRECEDING TARGET RECORD
*****

```

```

LOGIC -
VIEWATTN GIVES AN APPROXIMATE POINT AT WHICH TO START THE
SEARCH. BACKUP FROM THIS POINT UNTIL WE FIND THE FIRST
POINT NOT AFTER THE TARGET RECORD. THEN MOVE FORWARD UNTIL
WE FIND THE FIRST WHICH POINTS BEYOND THE DESIRED RECORD.
THEN USE THE ONE IMMEDIATELY PREVIOUS TO THAT. IN MOST CASES,
EXCEPT WHEN THE VIEW IS RADICALLY CHANGED, THIS OPERATION
WILL ONLY REQUIRE TWO COMPARES.

```

```

      R4,VIEWATTN
      RESUME PMM ASSOCIATED WITH THIS VIEW

```

```

LOOPBACK
DS
LTR R4,R4
BMP FROMBEGN
SPACE 1
C P3,PHEFILE(R4)
ERROR NP
SPACE 1
CALL

```

```

      DATACOMP,V1=NO,M2=(E,GATHWORK)
      FILEDEPN
      PRECEID(R4),
      GATHRECI
      R15,R15
      R4,R4
      ROTR
      GOFORWED
      SPACE 1

```

```

**** CONTINUE BACKING UP THE POINTER
      R4,VIEWATTN+4(R4)
      LOOPBACK
      BACKUP TO POSSIBLE PREVIOUS POINTER
      CONTINUE PROCESSING

```

0278722

33 / 46

81

PAGE 003

FILE: GATHFONT COPY A1 FISCHER-INNIS SYSTEMS CORP.

```

SPACE 1
***** START SEARCH FROM BEGINNING OF FILE'S ATTN-PHE LIST *****
FROMBECH DS
OR
R4, FILEATTN
-> FIRST ATTN PHE
LTR R2, R4
DO ANY EXIST?
BNP SETNULL
NO - THERE ARE NO ATTN ELEMENTS
SPACE 1
*****
ADVANCE FORWARD THROUGH PHE-ATTN CHAIN UNTIL WE
PIND FIRST ITEM AT OR BEYOND TARGET RECORD.
*****
SPACE 1
*****
GOFORWARD DS
OR
R2, R4
R4, PHEATTN (R4)
BNP USEPREV
R3, PHEFILE (R4)
ERROR NE
SPACE 1
*****
* COMPARE LOGICAL RELATION THIS PHE WITH TARGET RECORD
CALL
DATA COMP, VL=NO, MP=(E, GATHWORK),
FILEDEFN
PHEPRECID (R4),
GATHREC1)
R15, R15
BN2 GATH0800
LTR R0, R0
BNH GOFORWARD
SPACE 1
*****
*** WE HAVE REACHED DECISION POINT
R2 -> LATEST PHE-ATTN PRECEDING TARGET RECORD
*****
USEPR2V DS
OR
R14, PHE (P2)
LA PHE, R14
-> TARGET PHE
USING PHE, R14
PHEFUNC, PNTPONTE
IS THIS FONT-RANGE END?
CLI SETNULL
YES - FONT THEREFORE NOT OPEN
BE SETNULL
PHEFUNC, PNTPONTS
ELSE SHOULD BE FONT-START
ERR0R NE
THIS IS INCORRECT TYPE
MVC 2, PONTATTN (2, R7), PHEPONTI
SAVE ASSOCIATED PONT ID
DROP R14
SPACE 1
*****
DETERMINE 'CONTROL' FONT THAT IS IN EFFECT
*****
*****
SPACE 1
*****
*** DETERMINE IF THIS FALLS WITHIN CONTROL RANGE
R2 = INDEX OF PHE TO USE FOR NEXT SEARCH
SETNULL DS
OR
R2, VIEWATTN
R15, TPUNAIN
SET RESUME POINTER
-> MAINAREA
L

```



0278722

35/46

FILE: GATHPONT COPY A1 FISCHER-INNIS SYSTEMS CORP.

LA R10, PHE(R4) -> ASSOCIATED PHE  
 USING PHE, R14  
 MWC 2\*PORTCTL(2,R7), PHEPONTI SHOW CONTROL FONT IS ACTIVE  
 DROP R14  
 SPACE 1  
 \* CONTROL AND ATTENTION FONTS HAVE BEEN ANALYZED  
 NOCONTROL DS OH SHOW ALL WENT WELL  
 SB R15, R15  
 SPACE 1  
 :ATH0800 DS OH  
 POPUP RETH=R15

FILE: MERGPONT COPY A! FISCHER-INNIS SYSTEMS CORP.

```
***** TITLE 'MERGFONT - MERGE FONT VECTOR INTO ITEMAREA' *****
***** M E R G F O N T *****
***** MERGE FONT VECTOR INTO ITEMAREA *****
*****
DESCRIPTION -
THIS ROUTINE MERGES FONTS IN A VECTOR TOGETHER TO
CREATE A COMPOSITE 'ITEMAREA' SPECIFICATION.
*****
INPUT -
R0 -> FONT VECTOR
R1 -> ITEMAREA (ATTRIBUTES MUST BE CLEARED
      BEFORE CALLING THIS ROUTINE)
R12 -> TEWA
*****
OUTPUT -
NO EXCEPTIONAL CONDITIONS.
*****
COPYRIGHT 1986, ADDISON FISCHER - ALL RIGHTS RESERVED.
*****
SPACE 1 DEFINE IMMEDIATE LOCAL WORKING AREA *****
***** PUSHSECT , *****
***** DSECT XL(L,PASOATTH) *****
***** DS CL(L,PASICHAN) *****
***** DS CL(L,PASITAL) *****
***** DS XL(L,PASIPORE) *****
***** DS XL(L,PASIBACK) *****
***** DS XL(L,PASIBOLD) *****
***** DS XL(L,PASISHLIT) *****
***** DS XL(L,PASIOUFLT) *****
***** EQU **MRGFAREA *****
***** MRGFLNGH EQU POPSECT *****
***** SPACE 1 *****
***** USING TEWA,R12 *****
***** PUSHDOWN BASE=R11,SAVE=YES,TYPE=STACK, LINKAGE *****
***** WORK=(MRGFLNGH,MRGFAREA,R10) *****
***** SPACE 1 *****
***** CONSTRUCT COLLAPSED COMPOSITE OF ALL FONT LEVELS *****
***** (BUILT FROM THE BOTTOM UP). *****
*****
XC MRGFAREA(MRGFLNGH),MRGFAREA INITIALIZE WORK AREA
LR P5,R1
USING ITEMAREA,R5
LR R15,R0
LA R3,FONTLVL
SPACE 1
OH
DS R14,0(R15)
LH R14,P0,R14
LTBR NEXTLEV
DS
SPACE 1
R14,=Y(FASELNGH) GET OFFSET TO FONT ATTR & SPEC ELEM
HE
*****
LOUP4387
*****
```

0278722

$$\frac{36}{46}$$

FILE: HERFONTP COPY A1 FISCHER-INNIS SYSTEMS CORP.

R14,TEMAFONV -> ASSOCIATED (PINNED) FONTS VECTOR  
 USING PASE,R14  
 SPACE 1  
 \*\*\*\*\*  
 CH R3= (FONTLVLS-FONTSPEC) IS THIS 'SPEC' LEVEL?  
 BNE NOSPEC  
 SPACE 1  
 \*\*\*\*\*  
 CONSIDER SPECIAL FORMS-HANDLING FIELDS  
 PASEPCTL IS THIS SPECIAL FORM-CTL PONT?  
 BZ +8 NO - NOTHING TO SAVE  
 STH R0,ITERFORX SAVE FORMS-FONT INDEX  
 SPACE 1  
 \*\*\*\*\*  
 CONSIDER FOOTNOTE ATTRIBUTE  
 PASEPCL,PASEFOOT IS THIS A FOOTNOTE?  
 BZ +8  
 OY ITEM?,ITEMFOOT SHOW THIS ITEM IS A FOOTNOTE  
 SPACE 1  
 \*\*\*\*\*  
 APPLY BOTH FINAL & EXTENDED FORM  
 CONSIDER BASIC 3270 HIGHLIGHTING ATTRIBUTE  
 NOSPEC DS OH  
 CLI PASEATTR,PASEUNSP IS THIS UNSPECIFIED?  
 BR +10 YES - MAKE NO CHANGE  
 MVC HGFBCASC,FASUATTR SAVE BASIC HIGHLIGHT ATTR  
 SPACE 1  
 \*\*\*\*\*  
 CONSIDER EXTENDED ATTRIBUTES:  
 SHOULD BE RESET CHARACTER SET SPECIFICATION?  
 BNY SANECHAR,X'PP'  
 CLI SANECHAR,C''  
 BE SANECHAR  
 MVC HGFBCSET,PASICHAF SET BASIC CHARACTER SET FAMILY NAME  
 SPACE 1  
 \*\*\*\*\*  
 CONSIDER FONT MODIFIER  
 SANECHAR DS OH  
 CLI PASITAL,PASEUNSP ITALICS SPECIFIED  
 BE +10  
 MVC HGFITAL,PASITAL CHANGE ITALICS ATTRIBUTE  
 SPACE 1  
 \*\*\*\*\*  
 CLI PASIBOLD,PASEUNSP ITALICS SPECIFIED  
 BE +10  
 MVC HGFBBOLD,PASIBOLD CHANGE BOLD-FACE ATTRIBUTE  
 SPACE 1  
 \*\*\*\*\*  
 CONSIDER FOREGROUND COLOR  
 PASIFORE,PASEUNSP IS FOREGROUND (CHAR) COLOR SPECIFIED?  
 CLI +10  
 BE  
 MVC HGFFORE,PASIFORE USE BETTER COLOR  
 SPACE 1  
 \*\*\*\*\*  
 CONSIDER BACKGROUND COLOR  
 PASIBACK,PASEUNSP IS BACKGROUND COLOR SPECIFIED?  
 CLI +10  
 BE  
 MVC HGFBACK,PASIBACK USE BETTER COLOR  
 SPACE 1  
 \*\*\*\*\*  
 CONSIDER EXTENDED HIGHLIGHTING  
 PASIHIT,PASEUNSP IS EXTENDED HIGHLIGHTING SPECIFIED?  
 CLI +10  
 BE

0278722

37/uk

FILE: MERGPONT COPY A1 FISCHER-INNIS SYSTEMS CORP.

```

MVC MRGXHJL,FAS1HLIT YES, SET NEW SPECIFICATION
SPACE 1
* CONSIDER FIELD OUTLINING
CLI FAS1OUTL,FASEUMSP IS EXTENDED OUTLINING SPECIFIED?
DE *+10
KVC MRGFOUTL,FAS1OUTL YES, SET NEW SPECIFICATION
DROP R14
SPACE 1
**** ADVANCE TO NEXT ATTRIBUTE LEVEL
NEXTLEVL DS ON
LA R15,2(E,15) ADVANCE TO NEXT LEVEL
BC R3,LOOP4387 LOOP UNTIL HIGHEST LEVEL IS DONE
SPACE 1
*****
* CONVERT COMPOSITE FONT TO 'ITEMAREA' FORMAT
*
*****
SPACE 1
**** CONVERT TO 3270 COLOR ATTRIBUTE FORMAT
MVC ITEMBACK,MRGFBACK FOREGROUND COLOR SPECIFICATION
CLI ITEMBACK,0 SELECTING DEFAULT BACKGROUND COLOR?
BE *+8 JUMP IF SO
OI ITEMBACK,X'F0' ELSE SET ACTUAL 3270 ATTR VALUE.
SPACE 1
* HANDLE FOREGROUND (CHARACTER) COLOR ATTRIBUTE
MVC ITEMFORE,MRGFFORE FOREGROUND COLOR SPECIFICATION
CLI ITEMFORE,0 SELECTING DEFAULT BACKGROUND COLOR?
BE *+8 JUMP IF SO
OI ITEMFORE,X'F0' ELSE SET ACTUAL 3270 ATTR VALUE.
* CHECKING IF THE COLORS MATCH IS DONE BY THE 3270 ATTRIBUTE
* GENERATOR. IF THEY DO MATCH OUT THE FOREGROUND (CHARACTER)
* COLOR IS CHANGED TO EITHER BLACK OR WHITE.
SPACE 1
**** CONVERT CHARACTER SET SPECIFICATION
* ----- COMBINE CHARACTER SET NAME WITH ITALICS AND BOLDFACE.
**** CONSTRUCT
SPACE 1
MVC ITEMATTR,MRGFYHLL GET EXTENDED HIGHLIGHTING
NI ITEMATTR,ITEMFLSH+ITEMREV+ITEMOLIN+ITEMBRIT
CLI MRGPBASC,FASEUMSP SELECTING NORMAL INTENSITY?
DE *+8 JUMP IF SO
OI ITEMATTR,ITEMHIGH INDICATE BASIC HIGH INTENSITY
SPACE 1
POPUP, RETURN TO CALLER

```

0278722

38/26

0278722

39/46

FILE: POINTDEF COPY A1 FISCHER-INNIS SYSTEMS CORP.

```

*****
***** TITLE 'POINTDEF - DEFINE POINT-FUNCTION VALUES' *****
***** POINT DEF *****
*****
***** DEFINE POINT-FUNCTION VALUES *****
*****
***** THIS DEFINES THE FUNCTION CODES WHICH CAN BE ASCRIBED *****
***** TO DATA-POINTS WITHIN THE MASTER/WORD SYSTEM. *****
*****
***** COPYRIGHT 1986, ADDISON FISCHER - ALL RIGHTS RESERVED. *****
***** SPACE 1 *****
*****
***** PNTSTART EQU 0 START-OF-RECORD (INTERNAL HANDLING) *****
***** PNTINAPT EQU 1 INSERT-AFTER-PREVIOUS : PNE (ONLY) *****
*****
***** INSERT-AFTER-PREVIOUS DATA USES PREVIOUS *****
***** FORMAT BEFORE ANY SPECIFICATIONS ARE CHANGED. *****
*****
***** THIS IS USED WHEN INSERTION IS REQUESTED WITH *****
***** THE CURSOR ON A FORCED ATTRIBUTE BYTE (I.E. AN *****
***** ATTRIBUTE BYTE WHICH IS NOT OCCUPYING A BLANK'S *****
***** POSITION) WHICH DELIMITS A FONT CHANGE. IN THIS *****
***** CASE, WE NEED THIS DIFFERENT TYPE OF INSERTION *****
***** TO DISTINGUISH WHICH FONT THE INSERTION IS TO USE. *****
***** USE 'INSERT BYTE' (PREVIOUS BYTE) IF THE CURSOR *****
***** IS ON THE ATTRIBUTE. USE 'INSERT BEFORE' (FOLLOWING *****
***** BYTE) IF THE CURSOR IS ON THE FOLLOWING BYTE. *****
***** IN ALL OTHER CASES THE AFFILIATION OF A BYTE WITH *****
***** ITS FIELDS IS UNIQUE, AND 'INSERT-BEFORE' IS USED. *****
*****
***** PNTPONT EQU 2 FONT-RANGE END *****
***** PNE - BASE, SPEC, HILITE FONT *****
***** PNE - CONTROL/ATTR *****
*****
***** PNTPMEND EQU 3 FORMAT-FIELD END *****
***** PNE - FORMAT *****
***** PNE (FORMAT IS STRETCHED) *****
*****
***** PNTPMFIST EQU 4 FORMAT-FIELD START *****
***** PNE - FORMAT *****
***** PNE (FORMAT IS STRETCHED) *****
*****
***** PNTPARAI EQU 5 PARAGRAPH INDENT *****
***** PNE - FORMAT (ONLY) *****
***** PNE - BASE, SPEC, HILITE FONT *****
***** PNTPOINTS EQU 6 FONT-RANGE START *****
***** PNE - CONTROL/ATTR *****
*****
***** PNTTARGT EQU 7 TARGET POINT *****
***** PNE (ONLY) *****
***** PNTPEXACT EQU 8 PRECISE-POINT MARKER *****
***** PNE - HISTORICAL & INSERT PTS. *****
***** PNE - NAMED LABELS *****
***** PNE (IGNORED; PNE VERSION USED) *****
*****
***** PNTINBEP EQU 9 INSERT-BEFORE-CURRENT *****
***** PNE (ONLY) *****
***** PNTINEND EQU 10 END OF ACTUAL DATA *****
***** PNE - FORMAT (ONLY) *****
***** PNTSPILB EQU 11 SOFTWARE TAB *****
***** PNE - FORMAT (ONLY) *****
***** PNTNULLS EQU 12 START-NULLS *****
***** PNE - FORMAT (ONLY) *****
***** PNE (ONLY) *****
***** PNTAPPRX EQU 13 APPROXIMATE-POINT *****
***** PNE - FORMAT (ONLY) *****
***** PNE (ONLY) *****
***** PNTFINIS EQU 14 END OF PROCESSING *****
***** PNE - INTERNAL HANDLING *****
***** PNTAPPND EQU 15 APPEND *****
***** PNE - (INTERNAL HANDLING) *****
***** PNE - HISTORICAL & INSERT PTS. *****
***** PNE - APPEND-ROUTINE *****
*****
***** PNTINBAY EQU 16 *****
***** THIS INDEX RETURNS TO 'APPEND-ROUTINE'. *****
***** MAXIMUM NUMBER OF FUNCTIONS DEFINED *****

```





```
***** TITLE 'FASE - FONT ATTRIBUTE SPECIFICATION ELEMENT' *****  
***** F A S E *****  
***** FONT ATTRIBUTE SPECIFICATION ELEMENT *****  
  
THIS SPECIFIES THE CHARACTERISTICS OF A FONT WHEN  
USED IN EACH POSSIBLE CONTEXT.  
  
THE POSSIBLE CONTEXTS ARE:  
1. BASIC CRT FORM - NO EXTENDED ATTRIBUTES (REVISABLE)  
2. EXTENDED CPT FORM (REVISABLE)  
3. LINE PRINTER FORM (UNREVISABLE)  
4. PAGE PRINTER (SCRIPT) FORM (UNREVISABLE BY EDITOR)  
  
COPYRIGHT 1986, ADDISON FISCHER - ALL RIGHTS RESERVED.  
  
SPACE 1  
DSECT CL8  
FASERHAY DS W  
FASERENDY DS W  
FASERNSP EQU X'FF' (UNSPEC)  
FASERWCRH EQU Y'00' (NORMAL)  
SPACE 1  
FASERPCCL DS X'80'  
FASERSTDI EQU X'80'  
FASERNVLS EQU Y'40'  
FASERPRI EQU X'01'  
SPACE 1  
FASERPCCL DS X'80'  
FASERSTDI EQU X'80'  
FASERNVLS EQU Y'40'  
FASERPRI EQU X'01'  
SPACE 1  
FASERPCCL DS X'80'  
FASERSTDI EQU X'80'  
FASERNVLS EQU Y'40'  
FASERPRI EQU X'01'  
SPACE 1  
FASERPCCL DS X'80'  
FASERSTDI EQU X'80'  
FASERNVLS EQU Y'40'  
FASERPRI EQU X'01'
```

0278722

42/42

FILE: PASE COPY A1 FISCHER-INNIS SYSTEMS CORP.

```
* BASIC DATA EDITING OPTIONS
PASPFWH EQU X'04'    NUMERIC INPUT
PASSELFT EQU X'02'    LEFT-JUSTIFIED
PASSEFRSH EQU X'01'   RIGHT-JUSTIFIED
SPACE 1
PASSEXYIT DS CL4      EXIT SUFFIX (BINARY ZEROS => NO EXIT)
PASSEPCCD DS PL2      BINARY CODE FOR THIS FIELD
                                (NUMBER WHICH IS PASSED TO THE EXIT)
*****
SPACE 1
*****
REVISABLE (BASIC) CRT FORM
*****
OXL1          ATTRIBUTES FOR BASIC CRT
DS OX'40'     CRT INTENSITY
                HIGH INTENSITY
*****
SPACE 1
*****
REVISABLE (EXTENDED) CRT FORM
*****
OXL14         ATTRIBUTES FOR EXTENDED CRT
DS XL2 (INDEX?) NAME OF BASE CHARACTER SET
                (BLANKS IF NONE, OR DEFAULT)
                CHARACTER SET MODIFIER
                ON => USP ITALICS
                BOLDFACE INDICATOR
                ON => USE BOLDFACE
                COLOR ATTRIBUTE FOR FOREGROUND
                COLOR ATTRIBUTE FOR BACKGROUND
                BLUE (X'F1')
                PED (X'F2')
                PINK (X'F3')
                GREEN (X'F4')
                TURNOISE (X'F5')
                YELLOW (X'F6')
                NEUTRAL (X'F7')
*****
HIGHLIGHTING CHARACTERISTICS ARE MUTUALLY EXCLUSIVE)
EXTENDED HIGHLIGHTING
BLINKING (X'F1')
REVISED VIDEO (X'F2')
UNREFRESH (X'F4')
INTENSIFY (X'F8')
*****
OUTLINING
UNDERLINE
RIGHT-HAND-SIZE VERTICAL
OVERFLINING
LEFT-HAND-SIDE VERTICAL
```

0278722

43/16

```

PILE: PASE COPY A1 FISCHER-INNIS SYSTEMS CORP.
PAS10ALL EQU SPACE 1 PUT BOX AROUND THE FIELD
*****
* LINE PRINTER (UNREVISABLE) FORM *****
*
*****
SPACE 1
PAS2A1PR DS OXL2 ATTRIBUTES FOR LINE PRINTER
PAS2D1NS DS X 1 DARKNESS DENSITY
PAS2DARK EQU 2 DOUBLE STRIKE
PAS2H3VY EQU 4 VERY DARK (QUAD STRIKE)
PAS2HLIT DS SPACE 1
PAS2H4DR EQU 1 HIGHLIGHTING
PAS2US02 EQU 2 UNDERSCORE SINGLE STRIKE
PAS2J304 EQU 4 VERY DARK UNDERSCORE (QUAD STRIKE)
*****
* PAGE PRINTER (SCRIPT) FINAL FORM *****
*
*****
SPACE 1
PAS3A1PR DS OXL12 ATTRIBUTES FOR PAGE PRINTER (SCRIPT)
PAS3N1AE DS C18 NAME OF THE FONT
PAS3ITAL DS X 1 (BLANKS => DEFAULT FONT)
PAS3ITALV DS X 1 CHARACTER SET MODIFIER
PAS3BOLD DS X 1 ON => USE ITALICS
PAS3BOLD EQU 1 BOLDFACE INDICATOR
PAS3HLIT DS X 1 ON => USE BOLDFACE
PAS3H4DR EQU 1 UNDERSCORE
PAS3P1S2 DS FL1 POINT SIZE (RAW NUMBER)
PAS3L3SH EQU *-PASE LENGTH OF FONT ATTR SPEC ELEMENT

```

FILE: PNTSPEC COPY A1 FISCHER-INNIS SYSTEMS CORP.

PAGE 001

0278722

44/26

```

*****
***** TITLE 'PNTSPEC - WP/2 PNTSPEC SPECIFICATION' *****
***** PNTSPEC *****
***** WP/2 PNTSPEC SPECIFICATION *****
*****
***** COPYRIGHT 1986, ADDISON FISCHER - ALL RIGHTS RESERVED. *****
*****
***** SPACE 1 *****
*****
***** PNTSPEC CONTROL CODES *****
*****
***** S - START FIELD (NO LEFT JUSTIFY) *****
***** L - START FIELD (LEFT JUSTIFY) *****
***** R - END FIELD (NO RIGHT JUSTIFY) *****
***** T - END FIELD (RIGHT JUSTIFY) *****
***** H - HARDWARE (FIELD) TAB POINT *****
***** I - PARAGRAPH INDENT/INIT POSITION *****
***** 2 - START OF NULL FILL FOR THIS FIELD *****
***** (X - START OF BLANK FILL - NOT USEFUL) *****
*****
***** FUTURE IDEAS(?): N - START OF NUMBERED FIELD *****
*****
***** PNTSPEC CONTROL CODES *****
*****
***** W - WORD WRAP FINAL FIELD (=> 'W') *****
***** A - AUTO SPILL FINAL FIELD (=> 'A') => 'W') *****
***** E - HYPERHATE FINAL FIELD *****
*****
***** RULES *****
*****
***** 1. FIELDS DO NOT OVERLAP. *****
***** 2. HYPERHATE SPILL WORD WRAP APPLY TO, AND ARE SET IN, *****
***** ONLY (AT MOST) THE FINAL FIELD OF THE PNTSPEC *****
***** 3. THE PARAGRAPH INDENT ('I'), MAY OCCUR EITHER BEFORE OR *****
***** AFTER THE 'S'/'L' OF THE ASSOCIATED FIELD. IF ABSENT *****
***** IT IS CONSIDERED EQUAL TO THE 'S'/'L'. THIS ITEM ONLY *****
***** APPLIES TO THE FIRST LINE OF EACH PARAGRAPH. *****
***** THE 'I', 'S'/'L' AND 'E' ARE ALSO TREATED AS TAB ('T') FIELDS. *****
*****
*****
***** PNTSPEC FOUNDATION *****
*****
***** PNTSPEC DSECT CL8 PNTSPEC' *****
***** PNTSPEC DC ***** SIGNATURE OF THIS BLOCK *****
***** PNTNEXT DS F RSM -> NEXT FORMAT BLOCK *****
***** PNTPREV DS F RSM -> PREVIOUS FORMAT SPECIFICATION *****
*****
***** PNTSIZE DS 1 *****
***** PNTPOFF DS H ***** LENGTH OF FORMAT LINE *****
***** PNTPNTCT DS H ***** (ALLOWED TO EXCEED FINAL FIELD). *****
***** PNTPNTCT DS H ***** OFFSET TO START OF POINT VECTOR *****
***** PNTPNTCT DS H ***** COUNT OF FORMAT POINTS *****
***** PNTPNTCT DS H ***** (INCLUDES DUMMY 'TERMINATOR' POINT) *****
***** PNTPNTCT DS H ***** POINT TO BE USED TO DISPLAY -PREFIX- *****
***** PNTPNTCT DS H ***** COUNT OF FIELD DEFINITIONS *****
***** PNTPNTCT DS CL2 ***** ID OF THIS FORMAT *****
***** PNTPNTCT DS CL2 ***** IDENTIFIER OF OVERFLOW FORMAT *****

```

[illegible]

0278722

66 / 6.6

PAGE 003

FILE: PNTSPEC COPY A1 FISCHER-JENNIS SYSTEMS CORP.

PNTPARAI I PARAGRAPH INIT/INDENTATION POSITION  
PNTSTAB T SOFTWARE TAB-MARKER  
PNTNULLS Z/B START NULL/BLANK? FILL  
PNTFINIS TERMINATOR FOR END OF FORMAT

SPACE ;

\*\*\* ADDITIONALLY QUALIFY THE TYPE SPECIFICATION  
INDEX OF ASSOCIATED FIELD  
PPNTFLOX DS (USED FOR SIZE R I)  
PPNTLNGB PQV \*\*PNTPOINT LENGTH OF THIS ENTRY

**THIS PAGE BLANK (USPTO)**